Varian

Medical Systems

TrueBeam™

Developer Mode

User's Manual

(Draft – Not For Further Distribution – March 2012)

Developer Mode is intended for non-clinical use only and is NOT cleared for use on humans

This page intentionally blank

Reminder: Developer Mode is intended for non-clinical use only and is NOT cleared for use on humans

This page intentionally blank

Table of Contents

Introduction 1
Support 2
Operational Overview 2
Access to Developer Mode 3
User Interface 3
Creating Varian XML Beams 3
The MU vs. Position Trajectory Model 3
Composing XML beams 6
The Varian TrueBeam XML Schema 7
The simplest example: Beam On 7
Introducing static positions and tolerances 10
Adding static fields 13
A Dynamic Beam 15
MLC Carriage Movements 17
Specifying Dose Rate and Velocity limits 17
The Time Element: Dose Rates, Axis Speeds and Beam Delivery Time 19
Velocity Limits 20
Imaging 21
The Imaging Parameters Section 21
Taking two MV and two kV images concurrent with beam 21
Imaging Modes 26
Using Dummy Control points 28
Imaging Arm Positions, Tolerances and Motions 32
Taking Continuous Images 34
Axis numbering 37
Taking Images Only 37
Gating 41
Validating an XML beam 48
Loading and delivering an XML Beam 48
Indirectly setting up the KV foils, kV filters and kV collimators 50

DRAFT

Limitations for kV foil, kV filter, kV collimator and Imaging Calibration 51
Activating the Gating Hardware 51
Overwriting Interlocks 52
Using a DICOM file as a starting point 52
Trajectory Records (Trajectory Logs) 54
Enabling Trajectory Logs 54
Limitations 55
APPENDIX A: The Varian TrueBeam XML Schema 58
APPENDIX B: Accessories List 78

DRAFT

Introduction

This document describes how to use the Developer Mode research capabilities of the Varian TrueBeam linear accelerator. Developer Mode uses virtually the same control system as the TrueBeam Clinical modes. However, Developer Mode also enables access to additional more advanced control features which are not typically accessible or used by the Clinical Modes.

The user interface for Developer Mode is patterned after Service Mode. Developer Mode is permitted access to the full set of capabilities that have been built into the TrueBeam control system. Unlike the Clinical Modes, which are typically driven by plans downloaded through a DICOM interface, Developer Mode is driven by **XML Beams** loaded from local storage or network on the TrueBeam control console workstation computer. XML Beams are essentially text scripts in XML format where a rich instruction set allows Developer Mode users to construct and deliver complex non-standard beams, imaging and gating. In other words, it is this ability of Developer Mode to load and deliver XML beams that gives Developer Mode users full access to the complete dynamic beam , imaging and gating capabilities of the TrueBeam linear accelerator.

While the user interface (UI) for Developer Mode has a very similar look to Service Mode, the Developer Mode UI does not allow modifications to any machine configuration or operational parameters which may affect the functionality of Clinical Modes. So, for example, it is *not* possible to access the various accelerator subcomponents (*e.g.* functions such as beam and motor tuning) in Developer Mode. This general philosophy of Developer Mode guarantees that the normal operation of Clinical Modes cannot be affected by Developer Mode activities.

The following is a broad description of the additional control system features that are accessible through developer Mode:

- Define and deliver dynamic beams with arbitrary motion trajectories in MU-Position space for precisely controlled, predetermined variable dose rate, and simultaneous motion on potentially all accelerator axes (*e.g.* pre-programmed variable beam dose rate with simultaneous motion on gantry and/or potentially all collimation axes and/or potentially all couch axes).

- Program the acquisition of a variety of single images and/or start and stop several continuous imaging sequences at any predefined arbitrary point in the above beam/motion trajectory.

- Enable the respiratory gating signal to suspend/resume beam progression and/or imaging.

As a reminder: Developer mode is intended strictly for non-clinical use. Developer Mode is NOT cleared for use on humans.

Support

Questions about True Beam Developer Mode can be sent to:

TrueBeamDeveloper@Varian.com

Operational Overview

In summary, the typical workflow in using Developer Mode consists of:

a) Using an editor to create XML documents conforming to the Varian TrueBeam XML Schema,
b) transferring the XML script files to the accelerator's control system
c) loading and running the XML script files through Developer Mode

The following figure shows the basic steps involved:

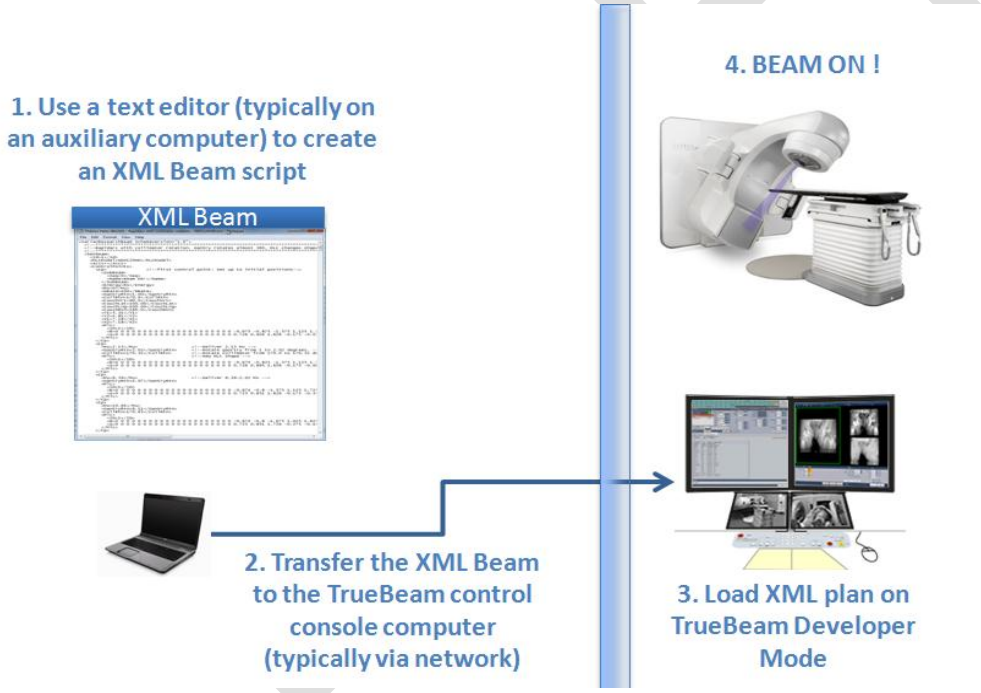

1. Use a text editor (typically on an auxiliary computer) to create an XML Beam script
XML Beam
2. Transfer the XML Beam to the TrueBeam control console computer (typically via network)
3. Load XML plan on TrueBeam Developer Mode
4. BEAM ON !

Fig 1. Elementary workflow of developing and delivering an XML beam

There are many methods of transferring the XML Beam file to the TrueBeam control console computer. However, transferring (copying) the file via network is recommended. Other methods, such as memory sticks, CDs and other digital media are more prone to infection with viruses and other malware. On TrueBeam accelerators, the I: drive is typically configured as a network drive and can be safely used to upload XML Beams to the accelerator.

In order to facilitate offline development, the typical workflow also includes two offline tools: an XML Beam Checker tool and an Emulator tool. These tools allow offline syntax checking and simulated delivery of XML Beams. These tools are described later in the document.

Access to Developer Mode

Access to Developer Mode is subject to license and user rights. Once a valid license has been issued, authorized users are allowed to launch Developer Mode by selecting "Research" from the TrueBeam control system main page and typing the user name and password.

Access rights for individual users are configured using the OSP Server platform as for Service Mode. It is recommended that users who need access to Developer Mode only (i.e. no acess to Service Mode) be given "Basic" user rights. The "Basic" level of access does not allow the overwriting of interlocks and faults which, in most cases, is adeguate for research purposes.

User Interface

As mentioned in the introduction, in terms of look and feel, the Developer Mode user interface is essentially a subset of the Service Mode user interface with one very powerful addition: The ability to load and deliver XML Beams.

Unlike Service Mode, those features that could affect machine performance in the Clinical Modes (e.g. Beam Tuning) have been disabled in Developer Mode. This is why, unlike Service Mode, Developer Mode does not provide low level access to the various accelerator sub-components. The purpose of this restriction is to keep Clinical Mode operations insulated from research performed using Developer Mode. The essential extra feature that Developer Mode provides compared to Service Mode is the ability to load and execute XML Beams.

Creating Varian XML Beams

Before we describe the details of how to create beam files using XML, it is important to first present the MU versus Position Trajectory Model used throughout the TrueBeam control system architecture.

The MU vs. Position Trajectory Model

The **MU versus Position Trajectory Model** or simply **Trajectory Model** is a central concept within the TrueBeam control system architecture and thus applies to all modes (Treatment, Service, Developer, QA etc.). According to the Trajectory Model, all beams, whether the typically more complex beams that are

enabled only through Developer Mode or the simpler beams available in the Clinical and Service Modes, are all described as a relationship between MU and position [1].

In more mathematical terms, the Trajectory Model treats the position of all mechanical axes as a function of MU. So there is one function per axis which describes movements (if any) during beam delivery, *i.e.* (Axis Position) = F (MU). These functions are called **Trajectory Functions** and the entire ensemble of trajectory functions for all axes is abstractly referred to as **The Trajectory**. For those axes that never move throughout a beam, the Trajectory Function degenerates to a flat line.

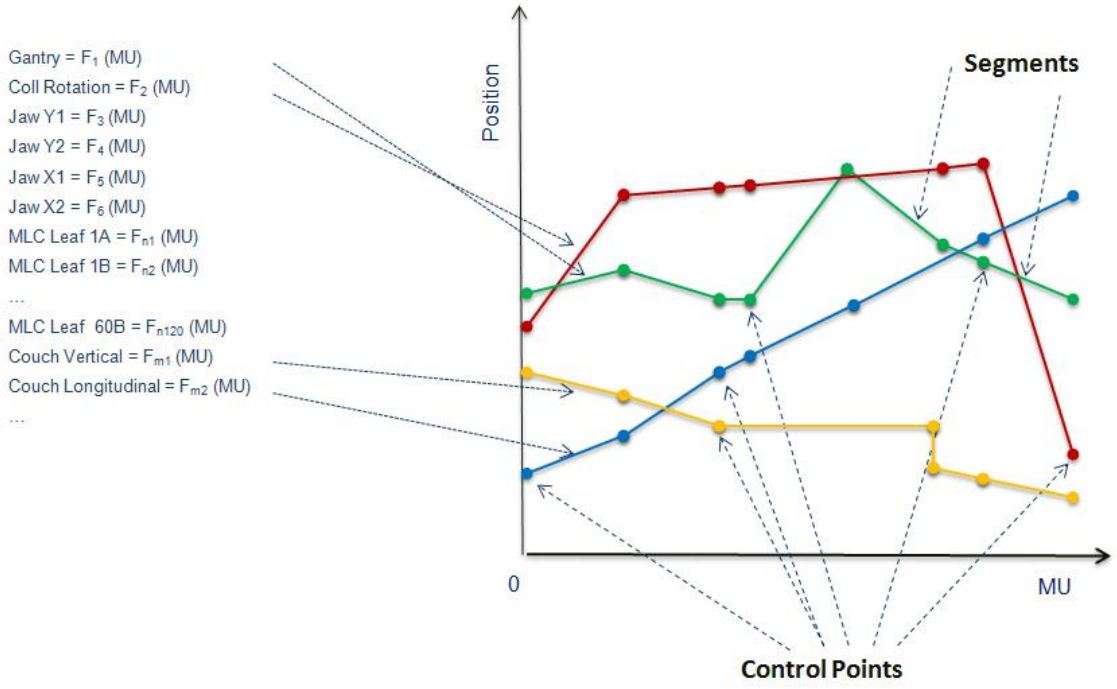


Fig 2. The MU vs. Position Trajectory model, Control Points and Segments

The Trajectory functions are segmented into linear segments. This facilitates representation of the function in digital form within the TrueBeam control system. In other words, the trajectory functions are described in terms of a finite number of discrete points called **Control Points**. The Developer Mode user (or any other planning entity for that matter) essentially programs the machine trajectory by specifying these Control Points in an XML Beam file.

[1] Note: The three imaging arms (MV imager, KV imager and KV source) can also be automatically moved during beams, however these arms do not follow the same exact dose vs. position model as the accelerator mechanical axes do. More details about moving the imaging arms can be found in the imaging section of this document.

The linear segment defined by two successive control points is simply called **Segment**. The piecewise-linear trajectory defined by the entire sequence of control points is referred to as the **Segmented Trajectory** or simply **Trajectory.** This piecewise-linear trajectory becomes the implied ideal trajectory to be followed at delivery time.

Another way to look at Control Points is that Control Points are essentially a mechanism to specify inflection points in the Trajectory *i.e.* points where the axis motion per MU delivered changes.

The sequencing convention for segments and control points is that the two successive control points *i-1* and *i*, define segment *i*. Therefore a beam starts with control point *0,* so control points *0* and *1* define segment *1*.

It is important to reiterate that there is an implied linear path within a segment and that the TrueBeam control system strictly enforces this implied linearity during beam delivery. The implied linearity is illustrated in the following example:

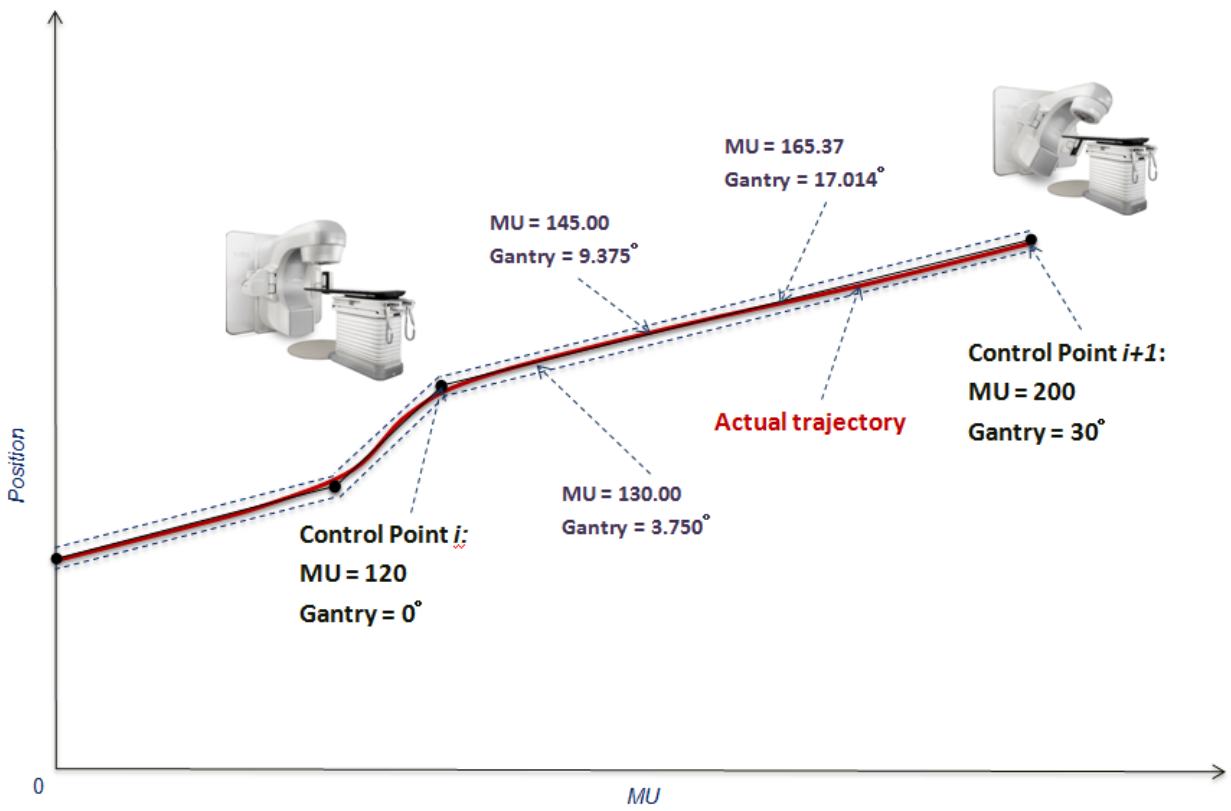


Fig 3. Trajectory and Linear Segment between Control Points.

The figure shows segment *i* defined by control points *i* and *i+1*.

By convention, beam delivery between control points is always specified in terms of cumulative MU. So, in the figure, segment *i* happens to define the arc like portion of a beam. During this arc segment, the gantry moves 30° (from position 0° to position 30°) while 80 MU (=200 MU- 120 MU) are delivered. The implied linearity means that the control system will control MU delivery and gantry angle so that the relationship between MU and gantry angle remains linear within any one segment. So, in the above example, when cumulative MU reaches 130 MU, the gantry will be at 3.750°, when the MU reaches 145 MU the gantry angle will be at 9.375° etc..

Notice that since the position of every mechanical axis is tied to MU, the relative position of axes with respect to each other is also precisely fixed during beam delivery. In other words, during beam delivery, motions follow an exact, predetermined, predictable and repeatable path, as specified in the XML Beam (the plan).

A segment that delivers radiation only without motion (a horizontal segment in the trajectory graph of figure 2) is referred to as a **Beam-Only Segment** or **Static Segment.**

A segment that simply moves one or more axes but delivers no MU (a vertical segment in the trajectory graph of figure 2, *i.e.* a segment defined by control points with the same cumulative MU) is referred to as a **Motion-Only Segment** or **Beamless Segment.** In principle, Motion-only segments are subject to the same standard linearity of motion principle, though there is a significant exception. The exception is that if a Motion only segment delivers neither MV nor KV radiation and there is MLC motion in the segment then the motions are not synchronized (see also "Motion-only segments may exhibit unsynchronized motion" in the "Limitations" section).

Composing XML beams

In principle, the only tool needed to use developer Mode is a text editor to compose XML files. There are also many commercially available XML editors that can simplify XML editing, but simple text editors are typically adequate.

As already mentioned, the basic operation in TrueBeam Developer Mode is to compose XML files and then load them on the TrueBeam control system to deliver beam and/or take images. These XML instruction files are referred to as **XML Beam Files** or simply **XML Beams** and, as is typical of xml files, they have an .xml extension.

XML is a widely used standard to transfer information between computers. XML is text, so it is easily read and manipulated using a text editor. XML does not do anything by itself and imposes only a few simple syntax rules. XML is simply a way to transfer information packets, called "elements" in XML parlance. Typically these information packets/elements are organized into a hierarchical tree structure, however, no particular tree structure is imposed by XML itself. It is essentially up to the entities that exchange information using the XML standard to define and agree upon the particular tree structure(s) that will be used, and the meaning of the information contained in the elements.

More, easy to read, general information on XML is provided at http://www.w3schools.com/xml/. The site provides simple tutorials about XML (Note: This site is not controlled or endorsed by Varian in any way. Similar sites can also be found through a simple web search).

The Varian TrueBeam XML Schema

For the specific case of the TrueBeam machine, the specific XML information tree accepted by the TrueBeam control system is defined in the so called **Varian TrueBeam XML Schema** or simply **TrueBeam Schema**. The TrueBeam schema specifies the valid syntax and acceptable information content of XML Beam files. Studying the TrueBeam schema is a way to learn about all the capabilities of Developer Mode.

Schema is yet another generic XML standard, a standard language used to describe schemas. It is also written in XML but its elements contain specific keywords used to describe the structure of XML trees; such as elements and syntax details about which child elements are mandatory, which are optional, the valid data types for an element etc.. A tutorial on XML schemas can be found at: http://www.w3schools.com/schema/default.asp .

Becoming familiar with the Varian XML Beam Schema is an essential part of doing research with Developer Mode. The Varian XML Beam Schema is described in the file **SetBeam.xsd**. The .xsd extension is typical of XML schema files. The **SetBeam.xsd** file is included in Appendix A. SetBeam.xsd can be used as a reference to the XML language accepted by TrueBeam. Varian also provides an electronic copy of SetBeam.xsd.

Upon examination of the Varian XML Beam schema, one sees that many of the elements in an XML Beam are optional, so the tree structure of XML beams can vary significantly from beam to beam. Simple fixed beams can be described as simple trees of just a few elements, while, in general, dynamic beams with more control points require more elements. Also, beams that take images are typically described by trees of greater depth since, in general, the information required to take images is more hierarchically organized.

The simplest example: Beam On

The following example is one of the simplest beams that can be written for the TrueBeam machine. When executed through developer mode, this beam simply delivers 100 monitor units of radiation.

```
<VarianResearchBeam SchemaVersion="1.0">
  <!--*********************************-->
  <!--Just beam on for 100 MU          -->
  <!--*********************************-->
  <SetBeam>
    <Id>1234</Id>
```

```
    <MLCModel>NDS120HD</MLCModel>
    <Accs></Accs>
    <ControlPoints>
      <Cp>
        <SubBeam>
          <Seq>0</Seq>
          <Name>Beam ON</Name>
        </SubBeam>
        <Energy>6x</Energy>
        <Mu>0</Mu>
        <DRate>300</DRate>
      </Cp>
      <Cp>
        <Mu>100</Mu>
      </Cp>
    </ControlPoints>
  </SetBeam>
</VarianResearchBeam>
```

Example 1: Just Beam On: One of the simplest XML beams.

The first element <VarianResearchBeam SchemaVersion="1.0"> is mandatory and the attribute SchemaVersion must currently be set to 1.0 as shown here[2]. The purpose of this attribute is to prevent incompatible files from being loaded on the TrueBeam machine (*e.g.* should a later version of the TrueBeam machine, with a different and incompatible XML syntax, become available sometime in the future).

The <SetBeam> element is the overall container element for the beam and is also required in every XML beam file.

The <Id> element is an integer identifier for the entire beam. Developer Mode does not impose any restriction on how this field is used. Essentially, Developer Mode users are free to identify their XML beams as they see fit.

The <MLCModel> is also required for every beam. It lists the MLC model intended to be used with this beam. For example, the standard Milleneum 120 MLC is referred to as “NDS120” whereas the smaller-leaf HD MLC is referred to as “NDS120HD”. If the specified MLC model does not match the MLC model on the target machine the control system will prevent beam-on through an interlock.

The <Accs> element lists any accessories that must be mounted to deliver this beam. A list of the accessories accepted by TrueBeam and their corresponding codes is listed in Appendix B. The control system will prevent beam, through an accessory interlock, until the listed accessories are mounted. Leaving this element empty (as in this example) means that any accessory (including no accessories) is compatible with this beam (*i.e.* accessories are essentially “don’t care” in this case). The element must always be present even if left empty.

[2] As of 2010. Check the Varian website for updated requirement(s).

<ControlPoints> is the overall container for the control points that will define the MU vs. Position Trajectory for this beam.

<Cp> is the container for parameters associated with a control point. Since control points are used to specify the segments of the Trajectory function, a beam must contain at least 2 control points (*i.e.* one segment minimum). The first control point is special as it must contain some additional parameters (see below).

The <SubBeam> element is mandatory at the first control point. However the SubBeam element is typically of little interest to Developer Mode users.[3] It contains an integer sequence identifier and a generic string name.

<Energy> is a mandatory element in the first control point. It specifies the MV beam energy to be used. The format is “dds" where:

dd = 0-99 and

s = 'x', 'e', or 'h', where,

x : MV X-Rays, e : MeV electrons, h : MeV HDTSe- electrons and

k : kV beams (used for beams that take kV images only and thus deliver no MV beam).

Examples: "6x" (for 6 MV X-Rays) "12e" (for 12 MeV electrons) and "0k" (for kV beam only).

The <Energy> element can appear only in the first control point. It is not possible to switch energies in the middle of an XML Beam. In other words, all MV radiation specified in a particular XML beam, is delivered at the same energy.

<Mu> is the cumulative MU for this control point. Since this is the first control point, the MU here must be zero.

<DRate> is the maximum dose rate to be used for this beam in Monitor Units per minute (300MU/min in this example). This element is mandatory in the first control point. For static beams the control system will deliver the Beam at this dose rate. However, for dynamic beams this dose rate is to be interpreted simply as a user defined “dose rate ceiling”, that is, a maximum dose rate not to be exceeded.

For dynamic beams where MU must be delivered concurrently with motions, the control system will generally compute and use a different dose rate on every segment. The segment dose rate used generally depends on which parameter is the limiting factor for that segment (*i.e.* MU or one of the concurrent motions). Therefore, in general, there is a maximum dose rate that can be used in a dynamic segment, depending on what else is moving in that same segment. If this maximum possible dose rate exceeds the ceiling dose rate specified by the user in the XML beam, then the control system will use the lower user-specified ceiling dose rate. Of course, the control system will then use also use

[3] The <SubBeam> is a construct used primarily by the clinical modes when radiation fields are combined into one automated field. However, since Developer Mode and the Clinical Modes use the same control system, this element is also present in Developer Mode.

proportionately lower speeds for the moving axes in the segment, in order to adhere to the MU vs. position Trajectory specified in the XML Beam.

It is important to reiterate that while the control system will never exceed the dose rate ceiling specified in the XML file, regardless of which one turns out to be the limiting factor on each segment (MU or one of the motions), the control system will always use an appropriate combination of dose rate and motion speeds so that the MU vs. Position Trajectory implied by the control points is satisfied. More information on his topic can be found in the section "The Time Element: Dose Rates, Axis Speeds and Beam Delivery Time" later in this document.

<Cp> is the container for the second control point. In this example, it contains only one element, <Mu>100</Mu> implying that 100MU will be delivered statically (*i.e.* radiation will be delivered while nothing moves).

In this particular example, all the positions of the mechanical axes, *i.e.* gantry, collimator rotation, MLC shape, couch positions etc. are not specified and are thus "don't care". This means that, for these axes, the control system will allow the user to beam on at any position.

Below are some additional, progressively more complicated XML beam examples. It is also possible to look at some of the XML beams that the TrueBeam control system itself generates. While Developer Mode cannot directly deliver DICOM plans, the Treatment modes convert every DICOM plan that is loaded into the control system into an XML Beam. Therefore, any DICOM plan that is deliverable through the TrueBeam Treatment mode can also be translated in its equivalent XML Beam format. For more details see the section "Using a DICOM file as a starting point".

Introducing static positions and tolerances

The following example adds enforcement of specific static positions and tolerances before the beam can be turned on.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--*************************************************************************-->
<!--200 MU at 400MU/min dose rate static field.                                -->
<!--Requires that some axes are positioned within tolerance prior to beam on.  -->
<!--*************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <TolTable>
      <GantryRtn>0.30</GantryRtn>
      <CouchVrt>2.00</CouchVrt>
      <CouchLng>1000.00</CouchLng>
      <CouchRtn>3.50</CouchRtn>
    </TolTable>
    <Accs></Accs>
```

```
    <ControlPoints>
      <Cp>
        <SubBeam>
          <Seq>0</Seq>
          <Name>Beam ON</Name>
        </SubBeam>
        <Energy>6x</Energy>
        <Mu>0</Mu>
        <DRate>400</DRate>
        <GantryRtn>221.00</GantryRtn>
        <CollRtn>180.0</CollRtn>
        <CouchVrt>90.54</CouchVrt>
        <CouchLng>100.00</CouchLng>
        <CouchRtn>180.0</CouchRtn>
        <Y1>5.31</Y1>
        <Y2>4.81</Y2>
        <X1>7.18</X1>
        <X2>7.18</X2>
        <Mlc>
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.675 0.875 1.575 1.125 1.725
             2.225 2.725 3.225 3.425 1.025 3.325 2.925 0.425 1.825 1.425 0.075 0.275
             0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.726 0.926 1.626 0.175 0.475
             0.475 0.075 0.525 0.625 0.325 1.025 1.825 2.025 1.625 1.025 0.525 0.326
             0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
      <Cp>
        <Mu>200</Mu>
      </Cp>
    </ControlPoints>
  </SetBeam>
</VarianResearchBeam>
```

Example 2: Adding static positions and tolerances.

This example delivers 200 MU statically. Nothing moves during beam delivery. However, this example enforces specific mechanical positions. By specifying positions for Gantry Rotation, Collimator Rotation, Couch Longitudinal, Couch Vertical, Couch Rotational, Collimator jaws Y1, Y2, X1, X2 and an MLC shape, this example will not allow beam on unless the mechanical axes listed are at their specified position.

Units: As a general rule, all positions in XML Beam files are always in Varian scale and the units are centimeters (cm) for translational axes and degrees for rotational axes. There is no limit to the number of decimal points that can be specified, however there is a practical positioning accuracy limit associated with the finite resolution and accuracy of the machine's position sensors.

DRAFT

If the position of an axis is not listed at a control point, it is assumed that it will not move and thus remain at the position established at some previous control point. This avoids repetition of positions for axes that do not move, and thus reduces the size of XML files required to specify a beam. This rule, however, does not apply for individual MLC leaves. If any of the MLC leaves must move at some control point then all the leaves must be included, even those that do not move. In other words, the MLC shape is always listed as a unit and the MLC shape can be omitted from a control point only when the entire MLC shape remains the same, *i.e.* none of the leaves move between the control points.

Note: While it is redundant to re-list across control points the same position for an axis that does not move, doing so does not render the beam invalid.

Another related rule is that if an axis is going to move at some point in the beam, then an initial position must be established for this axis at the first control point. This, of course, is true regardless of whether MU is simultaneously delivered while the axis is moving during beam delivery, or whether the motion is part of a beamless segment. As a corollary, if an axis is not listed in the first control point, then it cannot appear in any subsequent control points.

You may notice that the XML Beam of this example does not include a position for Couch Lateral. It is not necessary to list positions for all the axes. Some can be left as "don't cares", in which case the control system will not enforce any particular positioning for those axes prior to beam-on.

The above example also specifies tolerances for some of the mechanical axes. The control system will prevent beam on unless the axes are positioned to their listed positions within their specified tolerance. Therefore, in this example, the gantry would have to be positioned to within 0.3 degrees of its listed position of 221.00 degrees. The couch vertical would have to be positioned to within ±2.00cm of its specified 90.54 cm position (*i.e.* between 88.54 and 92.54 cm). The couch rotation would have to be positioned to within ±3.5 degrees from 180 (*i.e.* between 175.5 and 183.5 degrees).

Notice that collimator rotation has a position of 180.00 degrees listed but no tolerance has been specified for that axis. When no tolerance is specified for an axis, the default is that a tight internal control system tolerance applies by default. This tolerance is 0.1 cm (centimeters) for translational axes and 0.1 degrees for rotational axes. So, in this example, collimator rotation would have to be positioned at 180.00 ± 0.1 degrees for the system to allow beam on.

The Multileaf Collimator (MLC) must always be at the exact specified shape in order to beam on. In other words, the MLC always defaults to an internal tight tolerance. It is not possible to specify wider tolerances for the leaf positions.

In the above example, notice how the couch longitudinal tolerance has been set very wide. In essence that means that the control system will allow beam at any couch longitudinal position. The effect is similar to leaving the couch position unspecified as a don't care. However specifying a position, albeit with a large tolerance, provides a target position to easily drive to at delivery time (see automatic positioning of axes prior to beam on later in the document).

Notice that tolerances apply only to the initial position of static axes, that is, axes that do not move during beam delivery. For dynamic axes any specified tolerance becomes meaningless and the tight

internal control system tolerance is applied for any axis which will move during beam delivery. This is true regardless of whether the axes will move concurrent with the beam or will move within the context of a motion only segment.

Finally, the position of static axes is actively frozen during beam delivery. The position is then monitored to within the internal tight tolerance of 0.1 cm or 0.1°. In the unlikely event that a static axis should move during beam delivery, a MOTN (Motion) interlock will occur.

Note on MLC leaf format: The positions of the leaves in the XML Beam must be separated by exactly one blank space. While the Beam Checker Tool allows a variable number of spaces, some TrueBeam control system versions require exactly one single space.

Note on MLC carriage positions: MLC leaves are mounted on a movable carriage. If the MLC will move while delivering the XML Beam, the control system automatically computes the optimal position of the carriages based on the MLC shape, or sequence of shapes. The carriages are then automatically moved into position together with the leaves when the user loads the XML Beam and performs a “Go To”. If the sequence of MLC shapes requested is such that multiple carriage positions are required, then multiple carriage positions are automatically computed when the beam is loaded into the control system. Then, during Beam delivery, when the point is reached where a new carriage position is needed, then beam, as well as all other motions, are temporarily suspended while the MLC carriages are moved to their new positions.

Adding static fields

The following example, shows a Beam that delivers two static fields. The Beam-On button is pressed only once. After the MU for the first field is delivered (static segment), the motion to the next field happens automatically, without beam (motion only segment). Then upon reaching the second field, the beam comes on automatically.

While in this example the motion and the beam are not simultaneous, from a control system standpoint this is considered a dynamic beam, since, in a general sense, it requires the coordination of both beam and motion after the Beam-On button is pressed.

```
<VarianResearchBeam SchemaVersion="1.0">
  <!--*************************************************************************-->
  <!--Two static MLC fields delivered at two different gantry angles. First field
  delivers 110 MU, second field 72 MU    -->
  <!--*************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs></Accs>
    <ControlPoints>
```

```
      <Cp>              <!--first control point: Set up to initial positions-->
        <SubBeam> <Seq>0</Seq> <Name>Beam ON</Name> </SubBeam>
        <Energy>6x</Energy>
        <Mu>0</Mu>
        <DRate>400</DRate>
        <GantryRtn>180.00</GantryRtn> <CollRtn>180.0</CollRtn>
        <CouchVrt>90.0</CouchVrt> <CouchLat>100.00</CouchLat>
        <CouchLng>100.00</CouchLng> <CouchRtn>180.0</CouchRtn>
        <Y1>5.31</Y1> <Y2>4.81</Y2> <X1>7.18</X1> <X2>7.18</X2>
        <Mlc>
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.675 0.875 1.575 1.125 1.725
             2.225 2.725 3.225 3.425 1.025 3.325 2.925 0.425 1.825 1.425 0.075 0.275
             0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.726 0.926 1.626 0.175 0.475
             0.475 0.075 0.525 0.625 0.325 1.025 1.825 2.025 1.625 1.025 0.525 0.326
             0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>

      <!--second control point: Just deliver 110 MU without moving-->
      <Cp> <Mu>110</Mu></Cp>

      <!--third control point: No MU, just move gantry to new position and Mlc to
       new shape-->
      <Cp>
        <GantryRtn>90.00</GantryRtn>
        <Mlc>
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.988 0.235 0.775 0.995 1.475
             1.525 1.725 2.825 2.525 3.625 3.825 1.325 3.725 3.125 0.725 1.925 1.620
            -0.170 -0.550 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.218 0.115 0.826 0.826 1.666
             0.215 0.775 0.975 0.275 0.525 0.825 0.725 1.025 1.725 2.325 2.025 1.425
             0.720 0.964 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>

      <!--fourth control point: Just deliver 72 MU (=182-110) without moving
      anything-->
      <Cp> <Mu>182</Mu> </Cp>
    </ControlPoints>
  </SetBeam>
</VarianResearchBeam>
```

Example 3: Adding static fields with MLC.

A Dynamic Beam

Below is a dynamic beam example. This beam is similar to a RapidArc®-type Beam except that it also rotates the collimator during irradiation. The gantry makes a full clockwise revolution from 360 to 0 degrees while MU is being delivered. As seen in the control points parameters, gantry rotation is not proportional to MU, therefore the MU is not uniformly distributed along the arc (*i.e.* variable MU per degree). However, within a segment (*i.e.* between any two successive control points) the behavior is linear, *i.e.* MU delivered, gantry motion, and change in MLC shape are all proportional. As previously mentioned, this intra-segment linearity is actively enforced by the control system.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--*************************************************************************-->
<!--RapidArc with collimator rotation. Gantry rotates 360, MLC changes shape and also
collimator rotates.   -->
<!--*************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120</MLCModel>
    <Accs></Accs>
    <ControlPoints>
      <Cp>              <!--first control point: Set up to initial positions-->
        <SubBeam><Seq>0</Seq><Name>Beam ON</Name></SubBeam>
        <Energy>6x</Energy>
        <Mu>0</Mu>
        <DRate>400</DRate>
        <GantryRtn>360.00</GantryRtn><CollRtn>177.05</CollRtn>
        <CouchVrt>90.0</CouchVrt><CouchLat>100.00</CouchLat>
        <CouchLng>100.00</CouchLng><CouchRtn>180.0</CouchRtn>
        <Y1>5.31</Y1><Y2>4.81</Y2><X1>7.18</X1><X2>7.18</X2>
        <Mlc><ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.550 -0.875 -1.465 1.125
             1.835 2.125 2.555 3.625 3.775 1.225 4.325 2.925 1.325 1.725 1.525 -0.175
            -0.275 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.726 0.926 1.626 -0.175
            -0.575 -0.495 -0.175 0.725 0.825 0.465 1.025 1.825 2.025 1.625 1.025
             0.525 0.326 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
      <Cp>
        <!--In this next control point, MU is being delivered uniformly as the
        Gantry and Collimator rotate, while MLC changes shape. -->
        <Mu>2.13</Mu>               <!--Deliver 2.13 MU -->
        <GantryRtn>357.03</GantryRtn> <!--Rotate gantry from 360 to 357.03 deg. -->
        <CollRtn>177.54</CollRtn> <!--Rotate collimator from 177.05 to 177.54 deg-->
        <Mlc>          <!--and linearly change to following MLC shape -->
```

```
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.674 -0.825 -1.575 1.125
             1.725 2.225 2.670 3.200 3.425 1.025 3.325 2.925 0.425 1.825 1.425 -0.075
            -0.275 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.726 0.895 1.626 -0.175 –
             0.466 -0.475 -0.075 0.525 0.625 0.325 1.025 1.825 2.025 1.625 1.025
             0.525 0.326 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
      <Cp>
        <Mu>6.09</Mu>                    <!--Deliver 6.09-2.13 MU -->
        <GantryRtn>355.07</GantryRtn>
        <Mlc>
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.674 -0.8 -1.575 1.125 1.725
             2.225 2.625 3.125 3.325 1.425 3.225 2.925 0.818 1.825 0.925 -0.175 –
             0.174 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.725 0.851 1.626 -0.175 –
             0.375 -0.075 -0.075 0.525 0.725 -0.475 1.125 1.025 2.025 1.525 1.025
             0.525 0.225 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
      …

      …Several more control points here…
      …
      …
      <Cp>
        <Mu>590.88</Mu>
        <GantryRtn>0.00</GantryRtn> <!--completed 360 gantry revolution.-->
        <Mlc><ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1.275 -1.475 -1.375 0.825 –
             2.975 -4.074 -4.274 3.125 3.425 3.525 3.425 3.125 -1.375 -0.482 1.425
             0.325 -0.775 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.326 1.526 1.426 1.918 3.125
             4.125 4.325 -2.675 -2.875 -1.975 -2.275 -2.275 2.225 1.525 1.025 0.325
             0.826 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
    </ControlPoints>
  </SetBeam>
</VarianResearchBeam>
```

Example 4: A dynamic beam with simultaneous MU delivery and motions.

Note on imaging arms motions: An XML Beam can also specify motions for the three imaging arms (MV imager, kV Imager and kV Source). However, these mechanical axes cannot be moved while the MV beam is on and their motion does not follow the standard linear intra-segment trajectory pattern as the

other mechanical axes do. This restriction is described in more detail in the section "Imaging Arm Positions, Tolerances and Motions". When the imaging arms are moved, all other motions, as well as beam, are stopped while the arms move. Motions for the imaging arms is specified via the Imaging Parameters. Imaging parameters are described later in this document.

Note and limitation on the kV collimators (also known as kV blades), kV imaging foils and kV imaging filters: The current version of Developer Mode does not support motion of the kV collimators and/or changing of the kV imaging foils and/or kV imaging filters during beam delivery. Fixed positions for the kV collimators as well as a single set of kV imaging foils and kV imaging filters, to be used throughout the beam, can requested in the XML Beam. However, these settings must be indirectly established prior to beam-on. For more details on this limitation see the section "Indirectly Setting Up the kV Foils, kV Filters and kV Blades" later in this document.

MLC Carriage Movements

Specifying Dose Rate and Velocity limits

All the previous XML beam examples contain a Dose Rate specification. As already mentioned, in general, this dose rate should be interpreted as a limit, a ceiling. Therefore, the dose rate parameter is better referred to as a **Dose Rate Ceiling**. As the name implies this is a limit setting which enables the user to instruct the control system to never exceed the specified dose rate limit. As already explained, in general, it is not possible to specify an absolute fixed target dose rate for a dynamic beam. This is because, in general, for a dynamic beam, there are other instantaneous factors such as motions or image acquisitions that may indirectly impose a lower dose rate. See also the section "The Time Element: Dose Rates, Axis Speeds and Beam Delivery Time" later in this document.

Similarly it is possible to specify a **Velocity Table** inside an XML Beam. The Velocity Table contains velocity limits for the various mechanical axes. These, again, are simply velocity limit/ceiling settings which enable the user to instruct the control system to never exceed the specified velocities[4]. In general, the actual instantaneous velocity of an axis may be lower than the set limit. This is because the actual instantaneous velocity within a segment will generally depend on other motions taking place or beam being delivered in the same segment[5].

[4] From a conceptual standpoint, when dose is viewed simply as another axis, dose rate is similar to velocity and thus the dose rate limit/ceiling is similar to the velocity limits/ceilings. They are all simply limits on rate of change.

[5] Imaging may also have some effect on velocities, as beam progression may have to be throttled down or suspended during certain phases of the image acquisition process.

The following XML Beam portion shows how a gantry velocity limit of 2.5 degrees per second and a dose rate ceiling limit of 520 MU/min may be specified for the previous RapidArc® example:

```
<VarianResearchBeam SchemaVersion="1.0">
<!--**************************************************************************-->
<!--RapidArc with collimator rotation. Gantry rotates 360, MLC changes shape and also
collimator rotates.   -->
<!--**************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120</MLCModel>
    <VelTable>
      <GantryRtn> 2.50 </GantryRtn>
    </VelTable>
    <Accs></Accs>
    <ControlPoints>
      <Cp>              <!--first control point: Set up to initial positions-->
        <SubBeam><Seq>0</Seq><Name>Beam ON!</Name></SubBeam>
        <Energy>6x</Energy>
        <Mu>0</Mu>
        <DRate>520</DRate>
        <GantryRtn>360.00</GantryRtn><CollRtn>177.05</CollRtn>
        <CouchVrt>90.0</CouchVrt><CouchLat>100.00</CouchLat>
        <CouchLng>100.00</CouchLng><CouchRtn>180.0</CouchRtn>
        <Y1>5.31</Y1><Y2>4.81</Y2><X1>7.18</X1><X2>7.18</X2>
        <Mlc><ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.550 -0.875 -1.465 1.125
            1.835 2.125 2.555 3.625 3.775 1.225 4.325 2.925 1.325 1.725 1.525 -0.175
            -0.275 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.726 0.926 1.626 -0.175
            -0.575 -0.495 -0.175 0.725 0.825 0.465 1.025 1.825 2.025 1.625 1.025
            0.525 0.326 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
      …
      …
      …
```

Example 5: Specifying Dose Rate and Velocity Limits

As mentioned, the limit will ensure that gantry speed will not exceed 2.5 degrees per second. However, the actual instantaneous gantry speed chosen by the control system may be lower for those segments of the RapidArc® which may involve delivery of large number of MU, large MLC leaf movements or, in this case, a lot of collimator rotation.

The velocity limit is global within the beam. The velocity limit for an axis will apply regardless of whether MU is delivered concurrently with motion or whether the motion happens within the context of a motion only segment (*i.e.* beamless segment). Specifying velocity limits for axes that never move during the beam has no effect.

The Time Element: Dose Rates, Axis Speeds and Beam Delivery Time

The XML beams do not contain a time element. In other words, it is not possible for the user to specify the exact time progression and overall time required to deliver an XML beam.

This is because it would be quite complicated for users to evaluate all the trajectory and dose interdependencies present in a dynamic beam. It is thus the machine, which has knowledge of its limitations (eg. dose rate, speeds, accelerations) that is best suited to convert the dose position relationship implied in a specific XML beam into a Dose-Position-Time trajectory.

As mentioned, users can exercise some limited control in the timing by specifying dose rate ceilings and velocity limits for various axes in their XML beams. However, there is no direct user control of the time element. The XML beam remains to a large extent simply the specification of an MU vs. position trajectory. The time element is eventually computed by the control system when the XML beam is loaded for delivery.

In an abstract way, by introducing the time element, the control system takes the XML MU vs. position trajectory and converts it into an MU vs. time and position vs. time trajectory. This conversion takes place a-priori, that is, before beam on, when the XML beam is loaded into the control system (to be more exact when the "PREPARE" button is pressed on the TrueBeam machine's Control Console).

In general, the control system will compute the time progression so that the beam is delivered as fast as possible given

(a) the inherent machine limits (*e.g.* maximum possible dose rates, maximum possible speeds, maximum accelerations etc.)
(b) the user specified MU vs. position trajectory in the XML beam, and
(c) any additional user specified limits such as a dose rate ceiling and axis velocity limits.

The control system computes the dose rate and axis velocities on a per segment basis. For every segment, the control system analyzes the number of MU to be delivered and total axis movement for that segment. It then determines what is referred to as the **Limiting Rate** for that segment. The limiting rate is the rate that limits how fast a segment can be delivered. The limiting rate may result to be either the dose rate or the maximum velocity (motion rate) of an axis. These, in turn, may result to be either the inherent machine maximum dose rate and inherent machine axis velocities or the (lower) user specified Dose Rate Ceiling and/or velocity limits. The control system uses the maximum possible rate for the Limiting Rate and then computes proportionally the segment dose rate and/or segment velocity

of the other axes necessary to maintain the user specified MU vs. position trajectory as specified in the XML Beam file, while still respecting any user imposed dose rate or velocity limits.

Generally, the limiting rate can vary from segment to segment. So, for example, it may turn out that a segment where the limiting rate is gantry velocity may be followed by a segment where the limiting rate is dose rate or the maximum velocity of another mechanical axis, eg. one of the MLC leaves.

The control system also contains algorithms to more precisely control the finite accelerations of various mechanical axes. Since the accelerator's mechanical parts have mass and associated inertia, it is generally not possible to instantaneously change their speed at segment transitions (*i.e.* at control points). The control system employees various more sophisticated algorithms to handle these segment transitions, so that the user specified MU vs. position trajectory as specified in the XML beams is followed at all times, including segment transitions.

The total (wall clock) time to deliver a beam is simply the sum of the times required to deliver every segment. However, note that the time counter shown on the Developer Mode user interface does not account for time spent in beamless segments (*i.e.* motion only segments). Therefore, when delivering an XML Beam, the time shown in the counter will not match the wall clock time for those beams that contain one or more motion only segments (*i.e.* the counter will typically show less than wall clock time in these cases).

Velocity Limits

The following maximum velocities apply to any XML Beam that moves one or more axes dynamically. Unless, the XML beam itself specifies a lower velocity limit (ceiling) then the following values are used to calculate the position vs. time trajectory for all moving axes. As explained in the previous section, the actual instantaneous velocity observed at any point in the beam may be lower than the limits below or any lower limit specified by the user in the XML Beam itself, because of other constraints such as dose delivered or other moving axes.

Axis	Max Velocity
MLC Carriage	1.20 cm/sec
MLC Leaf	2.50 cm/sec
GantryRtn	6.00 deg/sec
Collimator Rtn	15.00 deg/sec
CouchVrt	2.00 cm/sec
CouchLat	4.00 cm/sec
CouchLng	8.00 cm/sec
CouchRtn	3.00 deg/sec
Coll X1	2.40 cm/sec
Coll X2	2.40 cm/sec
Coll Y1	2.40 cm/sec
Coll Y2	2.40 cm/sec

DRAFT

Imaging

The following sections describe how to add imaging to the XML Beam files.

Kilo-Voltage and Mega-Voltage images can be taken anywhere in the XML beam. In the most general case, Kilo-Voltage (kV) and Mega-Voltage (MV) images are taken either before therapeutic MV MU is delivered and/or after therapeutic MV MU is delivered and/or interspersed into a beam while therapeutic MV MU is delivered or axes are moved in motion-only segments. It is also often desirable to construct XML Beams that take images only. We discuss the more general case first: XML Beams that take images concurrent with the MV therapeutic beam.

The Imaging Parameters Section

Imaging instructions inside an XML Beam are given via <ImagingPoints> which are logically equivalent to the <ControlPoints> that are used to specify the MU vs. position trajectory of the therapeutic MV beam. However, the <ImagingPoints> are specified after the <ControlPoints> in a separate section called <ImagingParameters>. The imaging points are then synchronized to the therapeutic beam by referencing the <ControlPoints> from inside the <ImagingPoints> (see following examples).

Taking two MV and two kV images concurrent with beam

The following example shows an XML Beam that delivers a 180 MU, 200 degree gantry rotation arc field. The Beam contains more than two control points to make the number MU per degree vary throughout the arc (*i.e.* inflection points are needed to describe non-linear behavior). Five regions of different MU per degree radiation density comprise the arc. Two MV images are taken concurrent with the beam, one at the 10 MU point and one at the 172.79 MU point. Also two kV images are taken, one at the 20 MU point and one at the 120 MU point.

While, generally speaking, the image acquisitions are, in a sense, concurrent with the MV beam, typically, image taking will interfere briefly with beam delivery. This is because the imager and beam delivery must "handshake"[6], as the imaging system must be notified that the imaging point has been reached, the MV beam must be suspended while reading images taken in certain modes etc. Typically, these handshake operations are short and only introduce minor delays into the XML beam delivery. In the following example, a very short pause (slowdown) is seen in both the gantry speed and the MV beam delivery while the 4 images are being acquired. However, these slowdowns are already planned by the control system a priori (*i.e.* before beam on). Therefore, in this example, the short gantry slowdown is synchronized with beam so that the implied linearity of MU vs. gantry angle (implied in the

[6] Portions of the handshake can be eliminated through the <handshake> parameters in the imaging section. However, the handshake cannot be completely eliminated even if the <handshake> parameter is set to "false".

arc) is maintained at all times. In other words, in this example, MU vs. position linearity in the arc segment is maintained at all times regardless of any imaging related temporary slowdowns.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--*****************************************************************************-->
<!--Arc Beam, delivers 180 MU over 200 degree angle.
    5 arc regions at different MU/deg.
    MV images taken at 10 MU and 172.79 MU
    KV images taken at 20 MU and 120 MU
-->
<!--*****************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs></Accs>
    <ControlPoints>
      <Cp> <!--control point 0 (zero).-->
        <SubBeam><Seq>0</Seq><Name>Arc N Image</Name></SubBeam>
        <Energy>6x</Energy><Mu>0</Mu><DRate>300</DRate><GantryRtn>0.00</GantryRtn>
      </Cp>
      <!--Control points 1,2,3,4,5 follow...-->
      <Cp> <Mu>40</Mu> <GantryRtn>40.00</GantryRtn> </Cp> <!--40 deg @1    MU/deg.-->
      <Cp> <Mu>100</Mu><GantryRtn>80.00</GantryRtn> </Cp> <!--40 deg @1.5  MU/deg.-->
      <Cp> <Mu>140</Mu><GantryRtn>100.00</GantryRtn></Cp> <!--20 deg @2.0  MU/deg.-->
      <Cp> <Mu>170</Mu><GantryRtn>160.00</GantryRtn></Cp> <!--60 deg @0.5  MU/deg.-->
      <Cp> <Mu>180</Mu><GantryRtn>200.00</GantryRtn></Cp> <!--40 deg @0.25 MU/deg.-->
    </ControlPoints>

    <!--This is the imaging instructions section of the beam.-->
    <ImagingParameters>
      <DuringTreatment />
      <ImagingPoints>
        <!--*********************************-->
        <!--Instructions for first MV image. -->
        <!--*********************************-->
        <ImagingPoint>
          <Cp>0.25</Cp> <!--Index image, to be taken at 10 MU point.-->
          <Acquisition>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs /> <!--will use default specs-->
            <AcquisitionParameters>
              <ImageMode id="Highres" />
              <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
              <MV /> <!--this is an MV image.-->
            </AcquisitionParameters>
          </Acquisition>
```

```
        </ImagingPoint>

        <!--*********************************-->
        <!--Instructions for first KV image. -->
        <!--*********************************-->
        <ImagingPoint>
          <Cp>0.5</Cp><!--Index image, to be taken at 20 MU point.-->
          <Acquisition>
            <AcquisitionId>2</AcquisitionId>
            <AcquisitionSpecs>
              <Handshake>true</Handshake>
              <KV>true</KV>
            </AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="HighQuality">
              </ImageMode>
              <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
              <KV>
                <KiloVolts>80</KiloVolts>
                <MilliAmperes>50</MilliAmperes>
                <MilliSeconds>10</MilliSeconds>
                <eFluoroLevelControl>None</eFluoroLevelControl>
              </KV>
            </AcquisitionParameters>
          </Acquisition>
        </ImagingPoint>

        <!--*********************************-->
        <!--Instructions for second KV image. -->
        <!--*********************************-->
        <ImagingPoint>
          <Cp>2.5</Cp><!--Index image, to be taken at 120 MU point.-->
          <Acquisition>
            <AcquisitionId>3</AcquisitionId>
            <AcquisitionSpecs>
              <Handshake>true</Handshake>
              <KV>true</KV>
            </AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="HighQuality">
              </ImageMode>
              <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
              <KV>
                <KiloVolts>70</KiloVolts>
                <MilliAmperes>15</MilliAmperes>
                <MilliSeconds>10</MilliSeconds>
                <eFluoroLevelControl>None</eFluoroLevelControl>
```

DRAFT

```
            </KV>
          </AcquisitionParameters>
        </Acquisition>
      </ImagingPoint>

      <!--**********************************-->
      <!--Instructions for second MV image. -->
      <!--**********************************-->
      <ImagingPoint>
        <Cp>4.279</Cp> <!--Index image, to be taken at 172.79 MU point.-->
        <Acquisition>
          <AcquisitionId>4</AcquisitionId>
          <AcquisitionSpecs />
          <AcquisitionParameters>
            <ImageMode id="Highres" />
            <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
            <MV />
          </AcquisitionParameters>
        </Acquisition>
      </ImagingPoint>

    </ImagingPoints>
    <ImagingTolerances /> <!--mandatory, even if empty-->
  </ImagingParameters>

 </SetBeam>
</VarianResearchBeam>
```

Example 6: Acquiring kV and MV images concurrent with MV beam.

The <DuringTreatment /> element signifies that the images are part of an XML Beam that also delivers therapeutic MV radiation. **NOTE:** Some elements contain the word "treatment" because the same XML scripts are internally used by the control system to deliver beam in the clinical modes. However, as a reminder: **Developer Mode is intended for non-clinical use only and is NOT cleared for use on humans.**

Specifying image acquisition points:

Notice how, in general, the imaging points reference the control points in terms of fractional control points. So, the <Cp>0.25</Cp> element in the first MV image is what indexes the image acquisition to the MV beam. It essentially tells the control system to take the MV image 0.25 into the 1st segment, *i.e.* fraction 0.25 along the way from control point 0 to control point 1. This is the point where the MU reaches 10 MU. Similarly element <Cp>4.279</Cp> in the second MV image tells the control system to

take the image at fraction 0.279 into the 5th segment (*i.e.* transition from control point 4 to control point 5). At fractional control point 4.279 the MU will be 172.79 MU[7].

The first MV image:

The <AcquisitionId>1</AcquisitionId> is a user specified unique integer identifier for the image. Typically it is desirable to assign a unique identifier to each image acquisition within an XML Beam.

<AcquisitionSpecs /> is an element that enables the user to change the default acquisition specifications. In this example the default specifications for the imaging mode specified will be used, therefore the element is left empty.

The <ImageMode id="Highres" /> element specifies the image mode. In this case High Resolution (also referred to as High Quality). There are other MV imaging modes available, such as “Continuous”, “Dosimetry”, “Triggered” and “Lowres” (also referred to as Low Dose). These modes correspond to those available in the Treatment and Service Modes. The Treatment imaging modes are also described in the “TrueBeam Technical Reference Guide Volume 2: Imaging” document. See also the section “Imaging Modes” below.

<CalibrationSet>DefaultCalibrationSetId</CalibrationSet> signifies that the default calibration set will be used for this image acquisition.

The first kV image:

The first kV image is taken half way into the first segment as implied by the <Cp>0.5</Cp> element. That computes to the point when the MU counter reaches 20 MU.

The <Handshake>true</Handshake> element allows the image exposure and image reading to be synchronized with MV beam delivery. While it is possible to take images where MV radiation and image acquisition (either kV or MV) are not synchronized, it is generally recommended that this parameter be set to true.

The <KV>true</KV> element signifies that this is a kV image.

<ImageMode id="HighQuality"> is the kV imaging mode. In this case “High Quality”. There are other kV imaging modes available, such as “DynamicGain”, “DynamicGainFluoro”, “DynamicGainLowFramerate”, “DynamicGainLowFramerateFluoro”, “LowDose”, “Triggered” and “Fluoro”. These modes correspond to those available in the Treatment and Service Modes. The Treatment imaging modes are also described in the “TrueBeam Technical Reference Guide Volume 2: Imaging” document. See also the section “Imaging Modes” below.

<CalibrationSet>DefaultCalibrationSetId</CalibrationSet> means that the default calibration set will be used to acquire this image.

[7] Note that MU precision is typically limited to the quantization imposed by integral beam pulses. The MU per beam pulse varies by energy. As an example, for 6MV X-Rays, each pulse delivers approximately 0.03 MU. Therefore dose precision at 6X is quantized around 0.03 MU. For the FFF energy modes MU per pulse and the resulting quantization is coarser (up to ~0.11MU).

<KiloVolts>80</KiloVolts> , <MilliAmperes>50</MilliAmperes> and <MilliSeconds>10</MilliSeconds> are the Kilo Voltage, current (in mA) and duration (in milli-seconds) of the kV image exposure pulse.

The second kV image:

This image is acquired at <Cp>2.5</Cp> which corresponds to the point where MU reaches the 120 MU point.

Acquisition parameters differ from the first kV image. Here a pulse of 70 kilo-Volts, 15 mA and 10 milliseconds in duration is used to pulse the kV source.

The second MV image:

This image is acquired at control point 4.279 which corresponds to an MU of 172.79 MU.

Imaging Modes

As shown in the previous example, the *id* attribute of the *ImageMode* element is used to select the basic imaging mode for a particular image, or sequence of images. Of course, in Developer Mode, a single XML Beam may combine acquisition of many different images, each image, or image sequence, using a different imaging mode. The *id* attribute strings that must be specified to select the various imaging modes are shown in the following table. A brief description of the basic imaging mode is also shown:

DRAFT

KV Imaging Modes					
Imaging Mode	String (id=)	Acquisition Mode	Acquisition Technique	Resolution	Definition
Dynamic Gain	"DynamicGain"	Dynamic Gain	Single Image	1024 x 768	A basic mode intended for CBCT projections. In this mode each pixel automatically switches to a larger gain before reaching saturation, allowing a wider dynamic range.
Dynamic Gain Fluoro	"DynamicGainFluoro"		Continuous	1024 x 768	As above, but Images are continuously acquired at 11 frames per second.
Dynamic Gain Low Framerate	"DynamicGainLowFramerate"	Dynamic Gain Low Framerate	Single Image	1024 x 768	See *Dynamic Gain.*
Dynamic Gain Low Framerate Fluoro	"DynamicGainLowFramerateFluoro"		Continuous	1024 x 768	As above, except that images are continuously acquired at 7 frames per second.
High Quality	"HighQuality"	High Quality		2048 x 1536	Full pixel resolution used.
Low Dose	”LowDose”	Low Dose	Single Image	1024 x 768	2x2 pixels merged into one to achieve higher contrast, but lower resolution.
Triggered	"Triggered"		Single Image	1024 x 768	As above, but the images are triggered by respiratory gating.
Fluoro	“Fluoro”		Continuous	1024 x 768	Same as *Low Dose* but images continuously readout at 15 frames per second

DRAFT

MV Imaging Modes					
Imaging Mode	String (id=)	Acquisition Mode	Acquisition Technique	Resolution	Definition
Continuous	"Continuous"	Sync	Synchronized	1024 x 768	The MV imager readout is synchronized with the pause between pulses. Several lines are read at each beam pulse, until the entire image is read out.
Dosimetry	"Dosimetry"	NoSyncDRI	Unsynchronized	1024 x 768	MV imager continuously reads out accumulated dose independent of the beam pulse.
High Quality	"Highres"	High Resolution	Radshot	1024 x 768	MV imager reads out the entire frame with full resolution during beam hold.
Triggered	"Triggered"		Radshot		Same as above but image acquisition is triggered by gating device.
Low Dose	“Lowres”	Low Resolution	Radshot	512 x 384	One image acquired in higher contrast, lower resolution mode.

Imaging modes that use the above *Single Image* KV or *Radshot* MV Acquisition techniques should be used in conjunction with the <Acquisition> element, while the imaging modes that use the *Continuous* KV, *Synchronized* MV or *Unsynchronized* MV Acquisition techniques should be used in conjunction with the <AcquisitionStart> and <AcquisitionStop> elements. Continuous imaging is discussed later in this document.

----------end insert ----------

Using Dummy Control points

When taking images, it is often convenient to introduce extra dummy control points in an XML beam. These dummy control points can then be used as reference points to index image acquisition.

The following example illustrates the concept. It delivers two static fields, except that motion from the first field to the next is automatic. In addition, a pair of orthogonal kV+MV images is taken before each field.

A total of four dummy control points have been introduced to facilitate indexing where images are to be taken.

This example also illustrates how an MV and a kV image can be acquired at the same MU point. Also notice how the MLC is opened to a rectangular field in order acquire the MV image and then closed to the field shape before delivering the remainder of the field MU.

```
<VarianResearchBeam SchemaVersion="1.0">
  <!--*****************************************************************************
  Two static MLC fields. First delivers 110 MU (3.2 for image + 106.8)
  Second field delivers 72 MU (3.2 for image + 68.8)
  Orthogonal MV+KV images taken before each field
  *****************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs></Accs>
    <ControlPoints>
      <Cp>  <!--cp 0: Set up to initial positions, square MLC field-->
        <SubBeam><Seq>0</Seq><Name>Demo Dummy control points</Name></SubBeam>
        <Energy>6x</Energy><Mu>0</Mu><DRate>600</DRate>
        <GantryRtn>180.00</GantryRtn><CollRtn>180.0</CollRtn>
        <CouchVrt>90.0</CouchVrt><CouchLat>100.00</CouchLat>
        <CouchLng>100.00</CouchLng><CouchRtn>180.0</CouchRtn>
        <Y1>5.00</Y1><Y2>5.00</Y2><X1>6.00</X1><X2>6.00</X2>
        <Mlc>
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5.000 5.000 5.000 5.000 5.000 5.000
          5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 0 0 0 0 0 0 0
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
          <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5.000 5.000 5.000 5.000 5.000 5.000
          5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 0 0 0 0 0 0 0
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
        </Mlc>
      </Cp>
      <Cp></Cp> <!--cp 1: Use dummy cp to index KV image. -->
      <Cp></Cp> <!--cp 2: Use dummy cp to index MV image. -->
      <Cp><Mu>3.2</Mu></Cp>   <!--cp 3: Deliver 3.2 MU to square field to
                                                          take MV image. -->
      <Cp> <!--cp 4: Drive MLC and jaws to field shape.-->
        <Y1>5.31</Y1><Y2>4.81</Y2><X1>7.18</X1><X2>7.18</X2>
        <Mlc>
          <ID>1</ID>
          <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 
```

```
        <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.726 0.926 1.626 0.175 0.475 0.475
        0.075 0.525 0.625 0.325 1.025 1.825 2.025 1.625 1.025 0.525 0.326 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
      </Mlc>
    </Cp>
    <Cp><Mu>110</Mu></Cp> <!--cp 5: deliver remaining 106.8 MU-->

    <Cp><!--cp 6: No MU, just move to next field and ready MLC for
                                                        square MV image-->
      <GantryRtn>90.00</GantryRtn>
      <Y1>5.00</Y1><Y2>5.00</Y2><X1>6.00</X1><X2>6.00</X2>
      <Mlc>
        <ID>1</ID>
        <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5.000 5.000 5.000 5.000 5.000 5.000
        5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
        <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5.000 5.000 5.000 5.000 5.000 5.000
        5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 5.000 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
      </Mlc>
    </Cp> <!--cp x: Drive MLC and jaws to 2nd field shape.-->
    <Cp></Cp> <!--cp 7: Use dummy cp to index KV image. -->
    <Cp></Cp> <!--cp 8: Use dummy cp to index MV image. -->
    <Cp><Mu>113.2</Mu></Cp>   <!--cp 9: Deliver 3.2 MU to square field
                                                    to take MV image. -->
    <Cp> <!--cp 10:  Drive MLC and jaws to field shape.-->
      <Y1>5.31</Y1><Y2>4.81</Y2><X1>7.18</X1><X2>7.18</X2>
      <Mlc>
        <ID>1</ID>
        <B>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.775 2.375 2.075 2.075 0.125 0.325
        0.225 0.175 1.275 1.475 0.825 1.425 2.325 2.425 2.225 1.525 0.525 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</B>
        <A>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.826 2.426 2.126 2.126 0.025 0.825
        1.225 1.525 1.625 2.625 2.825 2.625 2.025 1.525 1.025 0.525 0.025 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</A>
      </Mlc>
    </Cp>
    <Cp><Mu>182</Mu></Cp><!--cp 11: deliver remaining 68.8 MU for
                                                          2nd field-->
  </ControlPoints>

  <!--____________________________________________-->
  <!--  I  M  A  G  I  N  G     S  E  C  T  I  O  N  -->
  <!--____________________________________________-->
  <ImagingParameters>
    <DuringTreatment />
    <ImagingPoints>
      <!--_____________________
      Take KV image (1st field)
      _____________________-->
      <ImagingPoint>
        <Cp>1</Cp>
        <Acquisition>
```

```
      <AcquisitionId>1</AcquisitionId>
      <AcquisitionSpecs>
        <Handshake>true</Handshake><KV>true</KV>
      </AcquisitionSpecs>
      <AcquisitionParameters>
        <ImageMode id="HighQuality" />
        <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
        <KV>
          <KiloVolts>80</KiloVolts><MilliAmperes>50</MilliAmperes>
          <MilliSeconds>10</MilliSeconds>
          <eFluoroLevelControl>None</eFluoroLevelControl>
        </KV>
      </AcquisitionParameters>
    </Acquisition>
  </ImagingPoint>
  <!--_____________________
  Take MV image (1st field)
  _____________________-->
  <ImagingPoint>
    <Cp>2</Cp>
    <Acquisition>
      <AcquisitionId>2</AcquisitionId>
      <AcquisitionSpecs><Handshake>true</Handshake></AcquisitionSpecs>
      <AcquisitionParameters>
        <ImageMode id="Highres" />
        <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
        <MV />
      </AcquisitionParameters>
    </Acquisition>
  </ImagingPoint>

  <!--_____________________
  Take KV image (2nd field)
  _____________________-->
  <ImagingPoint>
    <Cp>7</Cp>
    <Acquisition>
      <AcquisitionId>3</AcquisitionId>
      <AcquisitionSpecs>
        <Handshake>true</Handshake><KV>true</KV>
      </AcquisitionSpecs>
      <AcquisitionParameters>
        <ImageMode id="HighQuality" />
        <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
        <KV>
          <KiloVolts>80</KiloVolts><MilliAmperes>50</MilliAmperes>
          <MilliSeconds>10</MilliSeconds>
          <eFluoroLevelControl>None</eFluoroLevelControl>
        </KV>
      </AcquisitionParameters>
    </Acquisition>
  </ImagingPoint>
  <!--_____________________
```

DRAFT

```
        Take MV image (2nd field)
        _________________________-->
        <ImagingPoint>
          <Cp>8</Cp>
          <Acquisition>
            <AcquisitionId>4</AcquisitionId>
            <AcquisitionSpecs><Handshake>true</Handshake></AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="Highres" />
              <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
              <MV />
            </AcquisitionParameters>
          </Acquisition>
        </ImagingPoint>
      </ImagingPoints>
      <ImagingTolerances />
    </ImagingParameters>
  </SetBeam>
</VarianResearchBeam>
```

Example 7: Dummy control points facilitate specifying where exactly images are to be acquired.

One imaging programming style could be to specify a dummy control point for every imaging action. So a dummy control point in <ControlPoints> is created for every imaging action specified in the </ImagingPoints>. These dummy control points act as reference handles for the imaging section and the imaging section can be written to reference only the dummy control points.

Imaging Arm Positions, Tolerances and Motions

The previous examples, did not specify positions for the imager arms. It was assumed that the imaging arms would be placed at the proper imaging positions prior to beam on.

However, it is possible to specify explicit positions for one or more of the three imaging arms. These positions must be specified through the Imaging Points elements in the Imaging Section of the XML beam. It is also possible to specify tolerances associated with these arm positions.

Finally it is possible to move one or more of the three imaging arms during the beam. However, as previously mentioned, imaging arm motions do **NOT** follow the linear segment model as other motions do.

If a beam specifies motions for one or more of the imaging arms then:

- The imaging arms do not move between control points. Then
- When the beam reaches a control point where new arm positions are specified (in general a fractional control point), then, any beam and motion on the other axes stop. Then
- The imaging arms are moved to their new positions. Once the imaging arms have reached their new positions then,
- Any beam and motion on the other axes resumes.

The following figure illustrates how imaging arms move relative to beam and other (non imaging arm) axis motions:

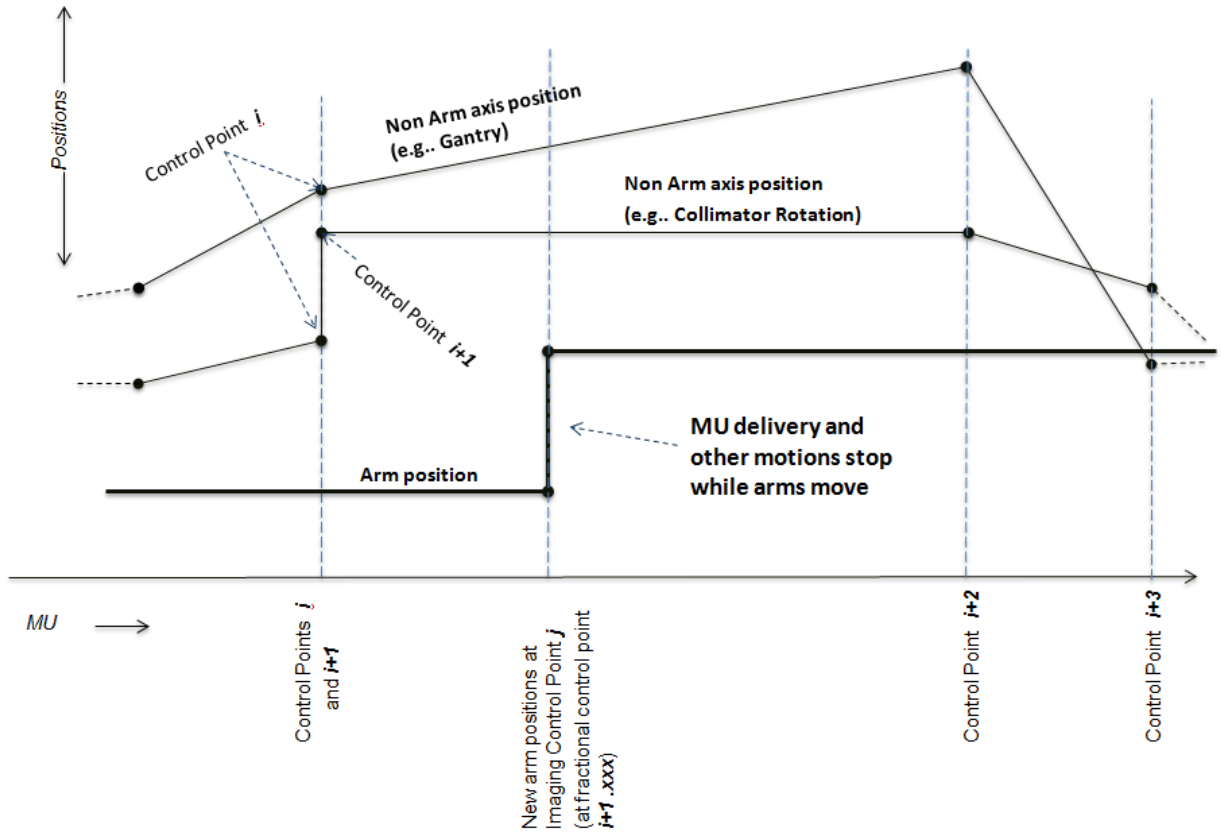


Fig. 4: Imaging Arm motions in MU vs. Position space

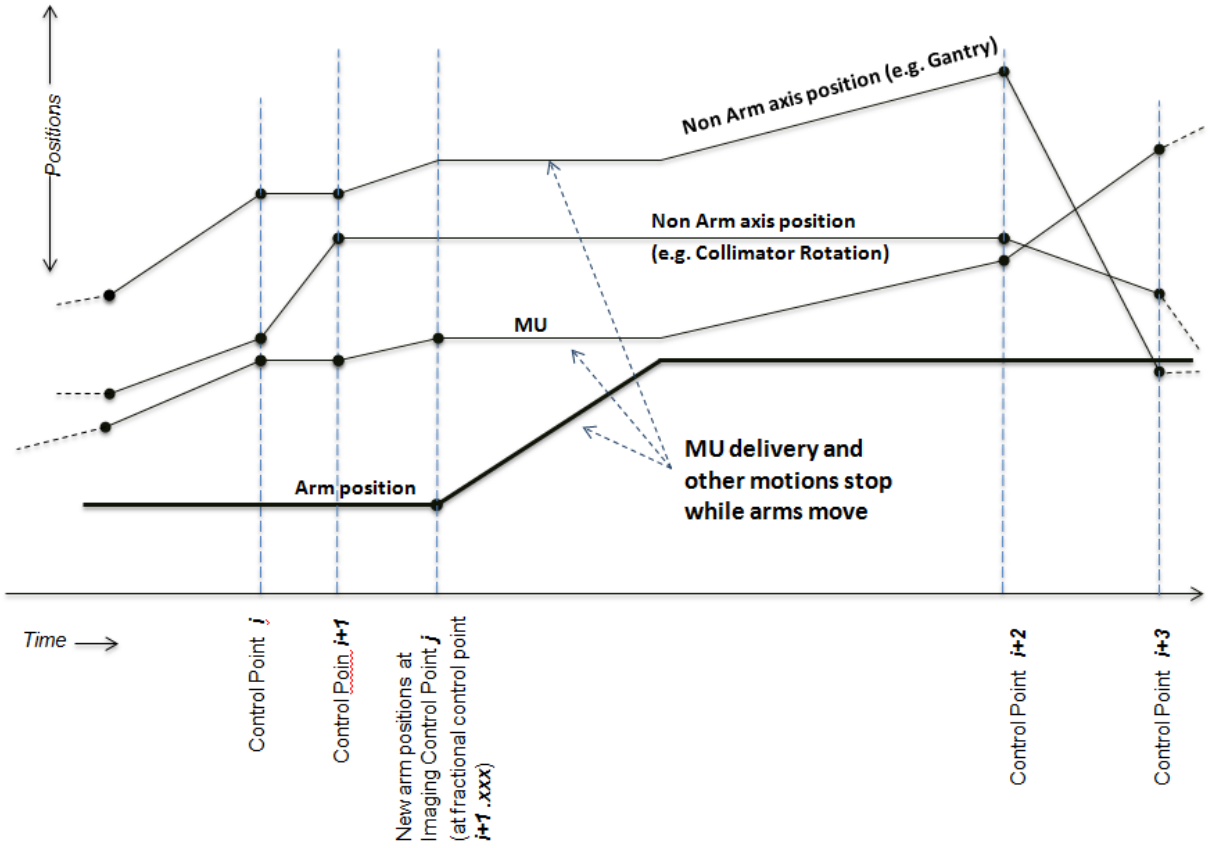


Fig. 5: Imaging Arm motions in Time vs. MU and Position space

As a corollary, it is not possible to move the imaging arms during irradiation.

The following example illustrates how to set imaging arm positions, tolerances and motions.

Xxxx Missing example: Better run this one on machine…

Show arc MV imager extended at start with tolerance. KVs come out and image for 90deg in middle of arc and then retract.

Example 8: Specifying Imaging Arm positions motions and tolerances.

Taking Continuous Images

Developer Mode also supports all the continuous imaging modes. Image acquisitions can be started and stopped at any point in the beam, using the Imaging Point elements in the Imaging Section of the XML Beam.

The following example illustrates continuous imaging. The example consists of a 180 to 0 degree arc beam where kV images are taken continuously in “DynamicGainFluoro” mode at 11 frames per second.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--*****************************************************************************-->
<!--Arc Beam, delivers 180 MU over 200 degree angle (0 to 200 deg).
    5 arc regions at different MU/deg.
    Continuous KV images are taken in the 40 to 90 degree region of the arc
-->
<!--*****************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs></Accs>
    <ControlPoints>
      <Cp> <!--control point 0 (zero).-->
        <SubBeam><Seq>0</Seq><Name>Arc N Image</Name></SubBeam>
        <Energy>6x</Energy><Mu>0</Mu><DRate>600</DRate><GantryRtn>0.00</GantryRtn>
      </Cp>
      <Cp> <Mu>40</Mu> <GantryRtn>40.00</GantryRtn> </Cp> <!--40 deg @1    MU/deg.-->
      <Cp> <Mu>100</Mu><GantryRtn>80.00</GantryRtn> </Cp> <!--40 deg @1.5  MU/deg.-->
      <Cp> <Mu>140</Mu><GantryRtn>100.00</GantryRtn></Cp> <!--20 deg @2.0  MU/deg.-->
      <Cp> <Mu>170</Mu><GantryRtn>160.00</GantryRtn></Cp> <!--60 deg @0.5  MU/deg.-->
      <Cp> <Mu>180</Mu><GantryRtn>200.00</GantryRtn></Cp> <!--40 deg @0.25 MU/deg.-->
    </ControlPoints>

    <!--This is the imaging instructions section of the beam.-->
    <ImagingParameters>
      <DuringTreatment />
      <ImagingPoints>
        <!--_____________________________________________________________________
        Make sure that KV source and imager are extended before starting
        _____________________________________________________________________-->
        <ImagingPoint>
          <Cp>0</Cp>
          <Kvd>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>-70</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvd>
          <Kvs>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>90</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvs>
        </ImagingPoint>

        <!--_____________________________________________________________________
```

```
        Start the KV fluoro acquisition at 40 degrees
        ____________________________________________________________________________-->
        <ImagingPoint>
          <Cp>1</Cp>
          <AcquisitionStart>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs><Handshake>true</Handshake></AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="DynamicGainFluoro"/>
              <CalibrationSet>IMG-CBCT</CalibrationSet>
              <ImageDestination>ImageDestination</ImageDestination>
              <KV>
                <KiloVolts>100</KiloVolts><MilliAmperes>20</MilliAmperes>
                <MilliSeconds>20</MilliSeconds><eFocalSpot>Large</eFocalSpot>
                <eFluoroLevelControl>None</eFluoroLevelControl>
              </KV>
            </AcquisitionParameters>
          </AcquisitionStart>
          <KvFilters><Foil>1</Foil><Shape>1</Shape></KvFilters>
          <KvBlades>
            <Positions>
              <KVX1>13.3</KVX1><KVX2>13.3</KVX2><KVY1>10.0</KVY1><KVY2>10.0</KVY2>
            </Positions>
          </KvBlades>
        </ImagingPoint>
        <!--____________________________________________________________
        Stop the KV fluoro acquisition at 90 degrees
        ____________________________________________________________-->
        <ImagingPoint>
          <Cp>2.5</Cp>
          <AcquisitionStop>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs />
          </AcquisitionStop>
        </ImagingPoint>
      </ImagingPoints>
      <ImagingTolerances />
    </ImagingParameters>
  </SetBeam>
</VarianResearchBeam>
```

Example 9: Taking continuous images.

If a continuous image acquisition is started but never stopped then it continues until the end of the beam at which point it is turned off automatically.

The line:

```
<ImageDestination>ImageDestination</ImageDestination>
```

Causes the control system to save evey single images it takes in continuous mode.

Axis numbering

Sometimes during control system operation error messages relating to a specific mechanical axis are displayed. These axes are often referred to by numerical index.

Following is a table of the numerical index convention for axes:

Index	Axis	Index	Axis	Index	Axis
1	Gantry Rotation	2	Collimator Rotation	3	Couch Vertical
4	Couch Lateral	5	Couch Longitudinal	6	Couch Rotation
7	Collimator X1	8	Collimator X2	9	Collimator Y1
10	Collimator Y2	11	Collimator MLC	12	Reserved
13	MV Detector Shoulder	14	MV Detector Elbow	15	MV Detector Wrist
16	MV Detector Hand	17	kV Detector Shoulder	18	kV Detector Elbow
19	kV Detector Wrist	20	kV Detector Hand	21	kV Source Shoulder
22	kV Source Elbow	23	kV Source Wrist	24	kV Source Hand
25	kV X1 Collimator	26	kV X2 Collimator	27	kV Y1 Collimator
28	kV Y2 Collimator	29	kV Filter Foil	30	kV Filter Shape

Taking Images Only

It is possible to construct XML Beams that take images only. This is done by specifying a special mode the <OutsideTreatment> mode in the imaging section of an XML beam.

The following example shows how the <outside treatment> mode can be used to take kV images only. This example rotates the gantry 368 degrees and sets up the kV imager to take continuous kV images in "DynamicGainFluoro" mode (11 fps). This type of beam could be used to acquire projections for Cone Beam CT (CBCT) reconstruction. Notice that no MV radiation is delivered in this Beam.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--******************************************************
Rotate gantry 368 degrees and take KV images to be used
for Cone Beam Reconstruction (CBCT).
******************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs/>
    <ControlPoints>
      <Cp>
```

```
        <SubBeam><Seq>0</Seq><Name>CBCT_FullFan360</Name></SubBeam>
        <Energy>0k</Energy>
        <!--*****************************************************
        MU and dose rate are 0 since we will be taking KV images only
        and we are using the outside treatment mode
        *****************************************************-->
        <Mu>0</Mu><DRate>0</DRate>
        <!--Start the gantry in the overtravel range. -->
        <GantryRtn>364</GantryRtn>
      </Cp>
      <Cp>
        <!--Gantry rotates 368 degrees all the way on the other side
         of the overtravel range -->
        <GantryRtn>-4</GantryRtn>
      </Cp>
    </ControlPoints>

    <!--Defines the imaging parameter for the beam and each control point -->
    <ImagingParameters>
      <OutsideTreatment>
        <!--*************************************************************
        No need to set a max MU because we're taking KV images only
        *************************************************************-->
        <MaxMu>0</MaxMu>

      <!--*************************************************************
      Here is where we specify that outside treatment mode
      *************************************************************-->
      </OutsideTreatment>
      <ImagingPoints>
        <!--*********************************************
        Start continuous acquisition in DynamicGainFluoro
        mode (a good choice for CBCT)
        *********************************************-->
        <ImagingPoint>
          <Cp>0</Cp>
          <AcquisitionStart>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs>
              <Handshake>true</Handshake>
              <KV>true</KV>
            </AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="DynamicGainFluoro"/>
              <CalibrationSet>IMG-CBCT</CalibrationSet>
              <ImageDestination>ImageDestination</ImageDestination>
              <KV>
```

```
                <KiloVolts>100</KiloVolts><MilliAmperes>40</MilliAmperes>
                <MilliSeconds>10</MilliSeconds><eFocalSpot>Large</eFocalSpot>
                <eFluoroLevelControl>None</eFluoroLevelControl>
              </KV>
            </AcquisitionParameters>
          </AcquisitionStart>
          <KvFilters><Foil>1</Foil><Shape>1</Shape></KvFilters>
          <Kvd>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>-50</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvd>
          <Kvs>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>100</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvs>
        </ImagingPoint>

        <!--*******************************************
        Stop the continuous image acquisition...
        *******************************************-->
        <ImagingPoint>
          <Cp>1</Cp>
          <AcquisitionStop>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs />
          </AcquisitionStop>
        </ImagingPoint>
      </ImagingPoints>
      <ImagingTolerances/>
    </ImagingParameters>
  </SetBeam>
</VarianResearchBeam>
```

Example 10: Taking images only (i.e. imaging only, no therapeutic beam)

One of the main advantages of the <OutsideTreatment> mode is that MV images can be acquired without having to explicitly specify the MU required to acquire these images. The control system automatically delivers the MV MU necessary to take the images based on the image type. In general, the MU necessary to take various images can be configured in Service Mode.

The following example illustrates how MV images can be acquired in the <outside treatment> mode leaving the number of MU unspecified and thus letting the control system deliver the minimum number of MU required for the requested images.

```
<VarianResearchBeam SchemaVersion="1.0">
```

```
<!--************************************************************
Takes MV image in outside treatment mode, thus leaving the MU
unspecified. Min MU required for image requested will be
pulled out of configuration for the image requested
************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs />
    <ControlPoints>
      <Cp>
        <SubBeam><Seq>0</Seq><Name>MV Outside</Name></SubBeam>
        <!--************************************************************
        Desired energy is set to 6MV...
        But MU is left at 0. Control system will deliver
        just enough MU to acquire the image type we are requesting.
        ************************************************************-->
        <Energy>6x</Energy>
        <Mu>0</Mu>
        <DRate>300</DRate>
      </Cp>
      <Cp />
    </ControlPoints>

    <ImagingParameters>

      <!--*********************************************************************
      Here is where we specify that outside treatment mode
      *********************************************************************-->
      <OutsideTreatment>
        <!--*********************************************************************
        This is just a limit of max MU to be delivered.
        Actual MU delivered will be just what is enough to take the MV image
        we are requesting.
        *********************************************************************-->
        <MaxMu>100</MaxMu>
      </OutsideTreatment>
      <ImagingPoints>
        <ImagingPoint>
          <Cp>0</Cp>
          <Acquisition>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs />
            <AcquisitionParameters>
              <ImageMode id="Highres" />
              <CalibrationSet>DefaultCalibrationSetId</CalibrationSet>
              <MV />
```

DRAFT

```
            </AcquisitionParameters>
          </Acquisition>
          <Mvd>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>-50</Vrt><Pitch>0</Pitch>
            </Positions>
          </Mvd>
        </ImagingPoint>
      </ImagingPoints>
      <ImagingTolerances />
    </ImagingParameters>

  </SetBeam>
</VarianResearchBeam>
```

Example 11: Letting the control system deliver only as much MV MU as necessary to acquire image(s)

Gating

It is possible specify XML Beams where the beam and/or imaging are gated in real time. The gating can be respiratory or, in general, some other device that provides real time gating information to the control system.

GATING THE kV Imaging beam:

Following is an example where kV imaging is gated. The example acquires images for CBCT but kV imaging stops when the gate is open. Notice that motions (*e.g.* gantry rotation in this example) is not gated. Only the kV beam is gated.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--*****************************************************
Rotate gantry 368 degrees and take KV images to be used
for Cone Beam Reconstruction (CBCT).
However take images only when gate is open and slow down
the gantry to take enough images for CBCT
*****************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>

    <!--*****************************************************
    Slow down the Gantry to take enough images for CBCT in spite of the gate
    *****************************************************-->
    <VelTable><GantryRtn> 2.00 </GantryRtn></VelTable>
```

```
    <Accs/>
    <ControlPoints>
      <Cp>
        <SubBeam><Seq>0</Seq><Name>CBCT_FullFan360Gated</Name></SubBeam>
        <Energy>0k</Energy>
        <Mu>0</Mu><DRate>0</DRate>
        <!--Start the gantry in the overtravel range. -->
        <GantryRtn>364</GantryRtn>
      </Cp>
      <Cp>
        <!--Gantry rotates 368 degrees all the way on the other side
         of the overtravel range -->
        <GantryRtn>-4</GantryRtn>
      </Cp>
    </ControlPoints>

    <!--Defines the imaging parameter for the beam and each control point -->
    <ImagingParameters>
      <OutsideTreatment>
        <MaxMu>0</MaxMu>
      </OutsideTreatment>
      <ImagingPoints>
        <!--**********************************************
        Start continuous acquisition in DynamicGainFluoro
        mode (a good choice for CBCT)
        **********************************************-->
        <ImagingPoint>
          <Cp>0</Cp>
          <AcquisitionStart>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs>
              <Handshake>true</Handshake>
              <KV>true</KV>
            </AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="DynamicGainFluoro"/>
              <CalibrationSet>IMG-CBCT</CalibrationSet>
              <ImageDestination>ImageDestination</ImageDestination>
              <KV>
                <KiloVolts>100</KiloVolts><MilliAmperes>40</MilliAmperes>
                <MilliSeconds>10</MilliSeconds><eFocalSpot>Large</eFocalSpot>
                <eFluoroLevelControl>None</eFluoroLevelControl>
              </KV>
            </AcquisitionParameters>
          </AcquisitionStart>
          <KvFilters><Foil>1</Foil><Shape>1</Shape></KvFilters>
          <Kvd>
```

DRAFT

```
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>-50</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvd>
          <Kvs>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>100</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvs>
        </ImagingPoint>

        <!--*******************************************
        Stop the continuous image acquisition...
        *******************************************-->
        <ImagingPoint>
          <Cp>1</Cp>
          <AcquisitionStop>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs />
          </AcquisitionStop>
        </ImagingPoint>
      </ImagingPoints>
      <ImagingTolerances/>


      <!--*******************************************
      Here are the gating parameters...
      *******************************************-->
      <GatingParameters>
        <Filter>BreathingPhase</Filter>
        <QualityThreshold>0.0</QualityThreshold>
        <!--*******************************************
        Setting KV gating to true here makes the KV obey the gate...
        *******************************************-->
        <KVGating>true</KVGating>
        <GatingWindow>
          <Entry>125</Entry> <!--Close gate starting at 125 degrees.-->
          <Exit>230</Exit>   <!--Open gate starting at 230 degrees. -->
        </GatingWindow>
      </GatingParameters>
    </ImagingParameters>

    <!--*******************************************
    The beam hold device must also be listed here in the
    beam hold device area.
    Beam hold device with id="4" is TrueBeam's stnadard
    respiratory gate.
```

DRAFT

```
    ********************************************-->
    <BeamHoldDevices>
      <Dev Id="4" />
    </BeamHoldDevices>

  </SetBeam>
</VarianResearchBeam>
```

Example 12: Gated kV images

The above example uses "Breathing Phase" gating with the gate closing between 125° and 230°. Other types of gating are also possible, such as amplitude or breath hold.

For amplitude gating the following XML excerpt can be used for the gating parameters:

```
…
<GatingParameters>
  <Filter>BreathingAmplitude</Filter>
  <QualityThreshold>0.01</QualityThreshold>
  <GatingWindow>
    <Entry>1.0</Entry>  <!--in centimeter units.-->
    <Exit>2.0</Exit>    <!--in centimeter units.-->
  </GatingWindow>
</GatingParameters>
…
```

For breath hold gating the following XML excerpt can be used for the gating parameters:

```
…
<GatingParameters>
  <Filter>PassThrough</Filter>
  <QualityThreshold>0.0</QualityThreshold>
  <GatingWindow>
    <Axis>z</Axis>
    <Entry>10</Entry>  <!--in centimeter units.-->
    <Exit>15</Exit>    <!--in centimeter units.-->
    <FaultOnExit>false</FaultOnExit>
    <EntryDelay>0.5</EntryDelay>
  </GatingWindow>
</GatingParameters>
…
```

GATING THE MV Imaging beam:

As mentioned, the MV beam can also be gated.

Notice that, when the MV beam is gated, all beam progression, *i.e.* both MV beam and any ongoing motions, are suspended while the gate is open. Typically this leads to those mechanical axes that were in motion while the gate opened, overshooting their position while they are aggressively stopped (overshoot is more likely to happen on the heavier axes, such as the gantry). However, the control system maintains active control of the mechanical axes and immediately drives back the overshooting axes to their appropriate resumption position. Then, once the gate closes again, the axes are in position ready to resume the beam. On resumption, the finite acceleration of the mechanical axes is taken into consideration and any beam is ramped up proportionately as to maintain the MU vs. position relationship specified in the XML beam at all times. If, following a gate open and overshoot, the mechanical axes have not returned to their resumption position by the time the gate closes again (*e.g.* axis overshoot during a very short gate) then beam remains suspended until all the overshooting axes return to their resumption positions.

Following is an example of a gated MV beam. The example is an arc with 5 different zones of MU/degree densities. kV images are being continuously taken in the 40 to 90 degree region of the arc. The MV beam, gantry rotation and kV imaging are all gated.

```
<VarianResearchBeam SchemaVersion="1.0">
<!--*****************************************************************************-->
<!--Arc Beam, delivers 180 MU over 200 degree angle (0 to 200 deg).
    5 arc regions at different MU/deg.
    Continuous KV images are taken in the 40 to 90 degree region of the arc
    However, both the MV beam as well as the imaging KV beam are gated
-->
<!--*****************************************************************************-->
  <SetBeam>
    <Id>1</Id>
    <MLCModel>NDS120HD</MLCModel>
    <Accs></Accs>
    <ControlPoints>
      <Cp> <!--control point 0 (zero).-->
        <SubBeam><Seq>0</Seq><Name>Arc N Image</Name></SubBeam>
        <Energy>6x</Energy><Mu>0</Mu><DRate>600</DRate><GantryRtn>0.00</GantryRtn>
      </Cp>
      <Cp> <Mu>40</Mu> <GantryRtn>40.00</GantryRtn> </Cp> <!--40 deg @1    MU/deg.-->
      <Cp> <Mu>100</Mu><GantryRtn>80.00</GantryRtn> </Cp> <!--40 deg @1.5  MU/deg.-->
      <Cp> <Mu>140</Mu><GantryRtn>100.00</GantryRtn></Cp> <!--20 deg @2.0  MU/deg.-->
      <Cp> <Mu>170</Mu><GantryRtn>160.00</GantryRtn></Cp> <!--60 deg @0.5  MU/deg.-->
      <Cp> <Mu>180</Mu><GantryRtn>200.00</GantryRtn></Cp> <!--40 deg @0.25 MU/deg.-->
    </ControlPoints>

    <!--This is the imaging instructions section of the beam.-->
    <ImagingParameters>
      <DuringTreatment />
```

```
      <ImagingPoints>
        <ImagingPoint>
          <Cp>0</Cp>
          <Kvd>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>-70</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvd>
          <Kvs>
            <Positions>
              <Lat>0</Lat><Lng>0</Lng><Vrt>90</Vrt><Pitch>0</Pitch>
            </Positions>
          </Kvs>
        </ImagingPoint>
        <!--_____________________________________________________________________
        Start the KV fluoro acquisition at 40 degrees
        _____________________________________________________________________-->
        <ImagingPoint>
          <Cp>1</Cp>
          <AcquisitionStart>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs><Handshake>true</Handshake></AcquisitionSpecs>
            <AcquisitionParameters>
              <ImageMode id="DynamicGainFluoro"/>
              <CalibrationSet>IMG-CBCT</CalibrationSet>
              <ImageDestination>ImageDestination</ImageDestination>
              <KV>
                <KiloVolts>100</KiloVolts><MilliAmperes>20</MilliAmperes>
                <MilliSeconds>20</MilliSeconds><eFocalSpot>Large</eFocalSpot>
                <eFluoroLevelControl>None</eFluoroLevelControl>
              </KV>
            </AcquisitionParameters>
          </AcquisitionStart>
          <!--Comment out foils and KV collimation for now...
          <KvFilters><Foil>1</Foil><Shape>1</Shape></KvFilters>
          <KvBlades>
            <Positions>
              <KVX1>13.3</KVX1><KVX2>13.3</KVX2><KVY1>10.0</KVY1><KVY2>10.0</KVY2>
            </Positions>
          </KvBlades>
          -->
        </ImagingPoint>
        <!--_____________________________________________________________
        Stop the KV fluoro acquisition at 90 degrees
        _____________________________________________________________-->
        <ImagingPoint>
          <Cp>2.5</Cp>
```

DRAFT

```
          <AcquisitionStop>
            <AcquisitionId>1</AcquisitionId>
            <AcquisitionSpecs />
          </AcquisitionStop>
        </ImagingPoint>
      </ImagingPoints>
      <ImagingTolerances />
      <!--*******************************************
      Here are the gating parameters...
      *******************************************-->
      <GatingParameters>
        <Filter>BreathingPhase</Filter>
        <QualityThreshold>0.0</QualityThreshold>
        <!--*******************************************
        Setting KV gating to true here makes the KV obey the gate...
        *******************************************-->
        <KVGating>true</KVGating>
        <GatingWindow>
          <Entry>125</Entry>
          <Exit>230</Exit>
        </GatingWindow>
      </GatingParameters>
    </ImagingParameters>

    <!--*******************************************
    The beam hold device must also be listed here in the
    beam hold device area.
    Beam hold device with id="4" is TrueBeam's stnadard
    respiratory gate.
    Establishing a gating device here, makes the MV beam
    obey the gate.
    *******************************************-->
    <BeamHoldDevices>
      <Dev Id="4" />
    </BeamHoldDevices>
  </SetBeam>
</VarianResearchBeam>
```

Example 12: Gating the MV beam.

Both the kV and MV beams can be gated in the same beam, however the gating parameters (gating window and gating thresholds) are common to both beams.

DRAFT

Validating an XML beam

For an XML Beam to be valid, it must conform to the TrueBeam XML Beam schema, i.e. it must be syntactically valid. It is typically convenient to check the syntactic validity of beams off-line, before attempting to load them into Developer Mode, as the TrueBeam accelerator is typically not available for experimentation at all times.

There are many simple commercially available XML editors and a host of other programs, such as Microsoft Excethat can be used to do this validation off-line on most PCs. While simple syntactic validation against the schema does not guarantee that the XML beam will be acceptable to the accelerator, the chance of simple syntax errors is notheless reduced. Many errors, such as, for example, the position of an axis specified in an XML beam being outside of the mechanical range, cannot be detected by simple schema validation, and thus the accelerator control system may still find further errors on an XML beam that has otherwise passed simple schema validation.

Loading and delivering an XML Beam

This section details how an XML beam is loaded into the TrueBeam control system and delivered (*i.e.* step 3 in figure 6).

To launch Developer Mode, select the “Research” option from the Major Mode Selection menu. Then type the appropriate User ID and Password.

Once in Developer Mode, you can load an XML Beam using the “Load Plan” button and selecting “Open Custom Beam” from the menu. A standard Windows file selection window will appear. Navigate to the folder containing the XML Beam you want to deliver and select “Open”. The XML Beam is now read into the TrueBeam control system. If the XML Beam is found to be invalid, an error will be displayed at this point.

Press the “PREPARE” button on the TrueBeam Control Console to prepare the beam for delivery. It is typically convenient to push the “Axis Positions” button at this point, to facilitate positioning the axes to their starting positions, as specified in the XML Beam. The initial positions required by the plan will appear in the “Program” columns of the interface. Pressing

"Go To" causes the initial axes positions from the XML Plan to become Target Positions. Target positions can now be activated in groups (Gantry & Collimator group, Couch Linear group and Couch Rotation group) and motion can be initiated using either the TrueBeam control console or the in-room pendants. The process is similar to activating Target Positions and moving axes in Treatment and Service Modes.

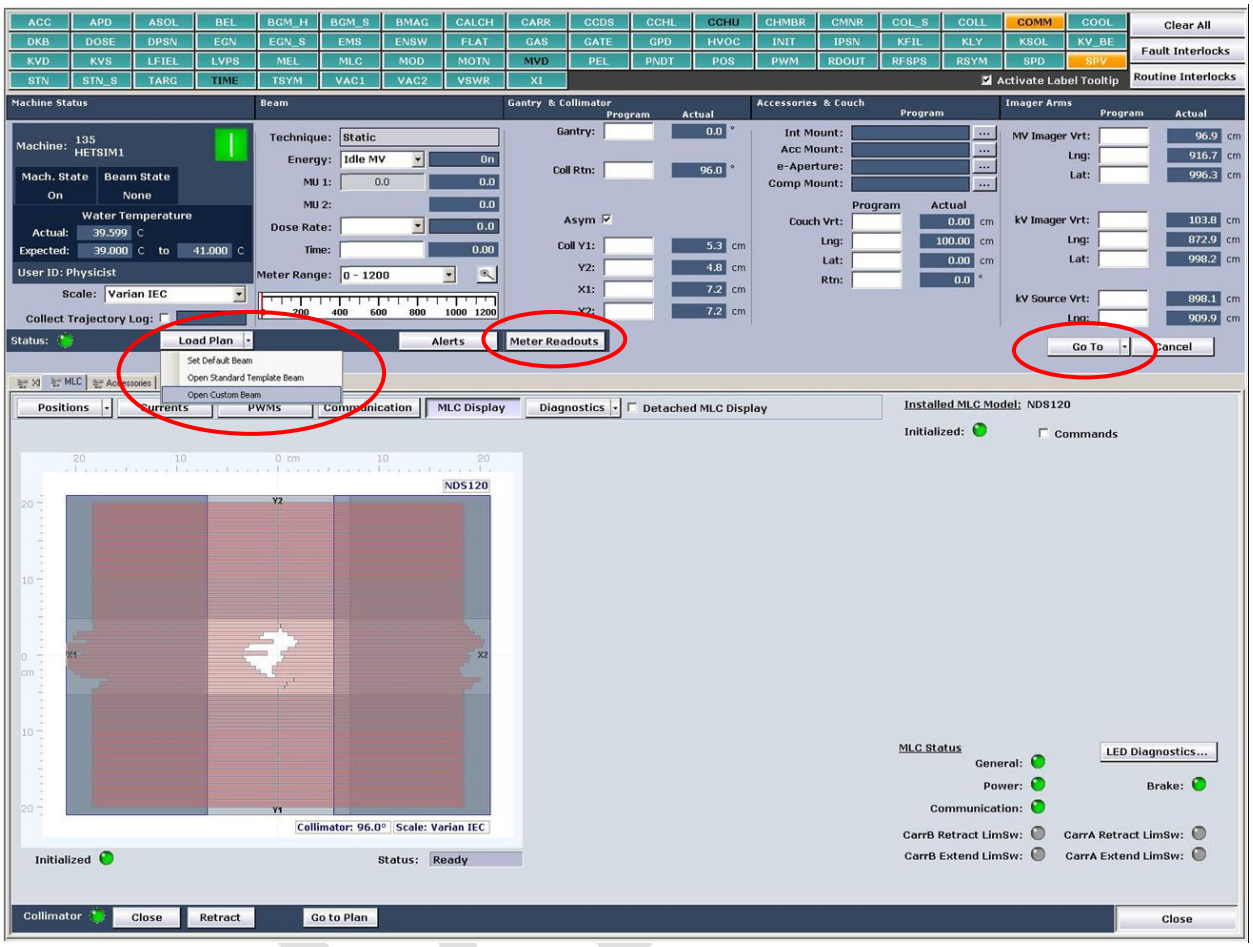
Note and Workaround: if your XML Beam requires either kV imaging foils and/or kV imaging filters, or if the XML Beam requires specific positions for the kV collimators, then you must use the trick described in the section "Indirectly Setting Up the kV Foils, kV Filters and kV Collimators" to activate these settings, since Developer Mode is currently unable to directly activate these settings prior to beam on. However, the control system still enforces these settings and will thus prevent Beam-On (through an "Initial Position Interlock") until the kV foils, kV filters and kV blades are set up to match any such settings requested in the XML Beam.

If the XML Beam specifies any collimator accessories, these accessories would have to mounted on the machine at this point. Otherwise an "Accessory" interlock will ultimately prevent beam-on.

Once all the axes have been positioned to their starting position the "MV Ready" button will light up on the Control Console. Proceeding to beam on is the same as for the Treatment Modes: Press and hold the "MV Ready" button for a few seconds. The system will go through a pre-beam-on check cycle, then the "MV Beam On" button will light. Pressing the "MV Beam On" button at this point initiates delivery of the XML Beam.

If the XML beam that was loaded contains only kV beam ("outside treatment") then only the "kV Beam on" button will light on the Control Console.

DRAFT



Indirectly setting up the KV foils, kV filters and kV collimators

Unfortunately, Developer Mode is currently unable to directly position the kV collimators (also known as kV blades), kV imaging foils and kV imaging filters. Therefore, if your XML Beam requests either kV imaging foils and/or kV imaging filters, or if the XML Beam requests specific positions for the kV collimators, then you must use the following process to indirectly set up those parameters prior to beam-on, using the imager calibration mode:

1. Exit Developer Mode and enter the Imager Calibration mode from the Major Mode Selection menu. Right click on a CBCT or other imaging mode and select “Acquire a Test Image” from the options (note: the system must not be in stand-by mode for this option to be available). Make sure to select a mode which will match the foil, shape filter and kV collimator positions in the XML Beam.

2. The system will immediately open a scan acquisition window and move the filter, foil and kV collimators to predetermined positions according to the imaging mode selected. Do not proceed with the calibration. Rather cancel out of the scan acquisition environment by selecting the green arrow in the upper left-hand corner of the Imager Calibration environment.

Remember that although the Developer Mode environment is currently unable to set up kV foils, kV filters and kV collimators, the system still interlocks if these settings do not match what is being requested in the XML beam.

Limitations for kV foil, kV filter, kV collimator and Imaging Calibration

The indirect procedure that must be used to set up kV foil, kV filter and kV collimator parameters, as described in the previous paragraph, implies that possible setups for these parameters in Developer Mode is limited to what is available in the Treatment modes.

Similarly imaging calibrations for CBCT acquisitions are limited to what is also available in the Treatment Modes. One implication is that CBCT acquisitions must be with the kV detector at 150 cm SID. This is not because of limitations in acquisition using an XML Beam. The reason is because calibrations are needed for reconstruction, and the only way to calibrate a CBCT mode is using the clinical system. Unfortunately, there is no way to modify the existing clinical modes so that a calibration can be made under non-standard conditions.

Activating the Gating Hardware

If your beam uses the gating hardware you must activate it before beam on. After you have pressed the "PREPARE" button on the TrueBeam Control Console , start the NDI camera acquisition manually by extending the tracking pane as shown in the image below.

NOTE: Tracking is not activated until you have pressed the "PREPARE" button on the TrueBeam Control Console.

Typically, approximately four tracking cycles are required to establish periodicity of the tracking signal (QualityThreshold). Once periodicity is established, the Analysis section on the display changes from orange into dark blue.

Overwriting Interlocks

With the appropriate user rights, it is possible for a developer mode user to overwrite machine interlocks. User rights in developer mode follow the same general principle as Service Mode.

In general, it is not recommended to overwrite interlocks in Developer Mode. It is expected that a functional machine will be able to deliver any valid XML beam without faults and interlocks. In that sense, it is recommended that users whose access to the TrueBeam machine is limited to Developer Mode be given "Basic" user rights in the OSP (Oncology System Platform) system. Developer Mode, like Service Mode, does not allow users with "Basic" rights to overwrite interlocks and faults.

Using a DICOM file as a starting point

It is possible to use the TrueBeam Treatment mode to convert a DICOM field (or even a sequence of fields) into a Beam XML. When a DICOM field is loaded into the Treatment Modes for delivery, the DICOM field is always first converted into the equivalent XML Beam, since XML Beam is the native TrueBeam control system format. It is therefore possible to exploit this fact and use the Treatment Mode to convert a DICOM file into an XML Beam.

Note: Converting Clinical DICOM fields into XML Beams and studying these system generated XML beams is perhaps an important step in learning how to compose XML Beams.

The system generated XML beams can then also be used as a starting point, further edited and modified to create new custom XML Beams.

The following is a description of how to use Clinical Mode to convert a DICOM field (or sequence of fields) into an XML Beam:

To perform the DICOM to Beam XML conversion, pick the field that you would like to convert and load it into the control system by pressing the "PREPARE" button on the Control Console. If you would like to convert an entire sequence of fields which are part of the same plan, use the normal field automation feature of TrueBeam (if you are unfamiliar with the field automation feature, see "TrueBeam TrueBeamSTx Instructions for Use" manual).

Once the field is loaded the c:\vmsos\temp\TDSTreatment\ LastSetBeam.xml file is written, so you can exit Treatment Mode (*i.e.* it is not necessary to proceed to actually delivering the beam).

Notice that the LastSetBeam.xml file as generated by the machine does not contain line breaks, so it is difficult to read and edit. Line breaks can be added by simply opening the file with a program, such as Microsoft XML Notepad and then saving it again (Microsoft XML Notepad can be downloaded as freeware). The new file containing line breaks can now be viewed and edited using any text editor, such as Microsoft Notepad.

Notice that the LastSetBeam.xml file thus generated cannot be directly delivered through Developer Mode. The following simple edits are necessary to make this file compatible for delivery in Developer Mode:

1. Add the tags <VarianResearchBeam SchemaVersion="1.0"> </VarianResearchBeam> so that they are the first and last tags in the entire file.
2. Remove the tags <TreatmentMode> </TreatmentMode> from the plan file including whatever tags are enclosed by the TreatmentMode tags. For example, <TreatmentMode><![CDATA[FTM]]></TreatmentMode> should be deleted.

Once these simple edits are performed, it should be possible to load and deliver the resulting XML beam through Developer Mode.

NOTE: Any gating parameters used in the DICOM file will not be included in the LastSetBeam.xml that is generated using the procedure above.

DRAFT

Trajectory Records (Trajectory Logs)

The TrueBeam control system employs many algorithms and active servo control mechanisms to ensure that the actual MU vs. position trajectory delivered matches the linear segmented trajectory specified by the user in the XML file (or other planning entity if in Treatment Mode). Generally, the control system will actively compute and manage MU delivery, velocity of moving parts, acceleration of moving parts and segment transitions, to match the actual trajectory with the planned segmented trajectory. So, for example, if MU delivery were to temporarily slow down somewhat, due to unexpected transient phenomena, the control system will proportionately slow down all moving axes so that beam progression continues along the planned segmented trajectory. When a beam first starts, (or every time the respiratory gate closes and beam resumes, if the beam is gated) the beam and motion of every moving axis are ramped up according to a pre-calculated acceleration profile. The objective is to accommodate the moving axis exhibiting the most demanding instantaneous inertia characteristics.

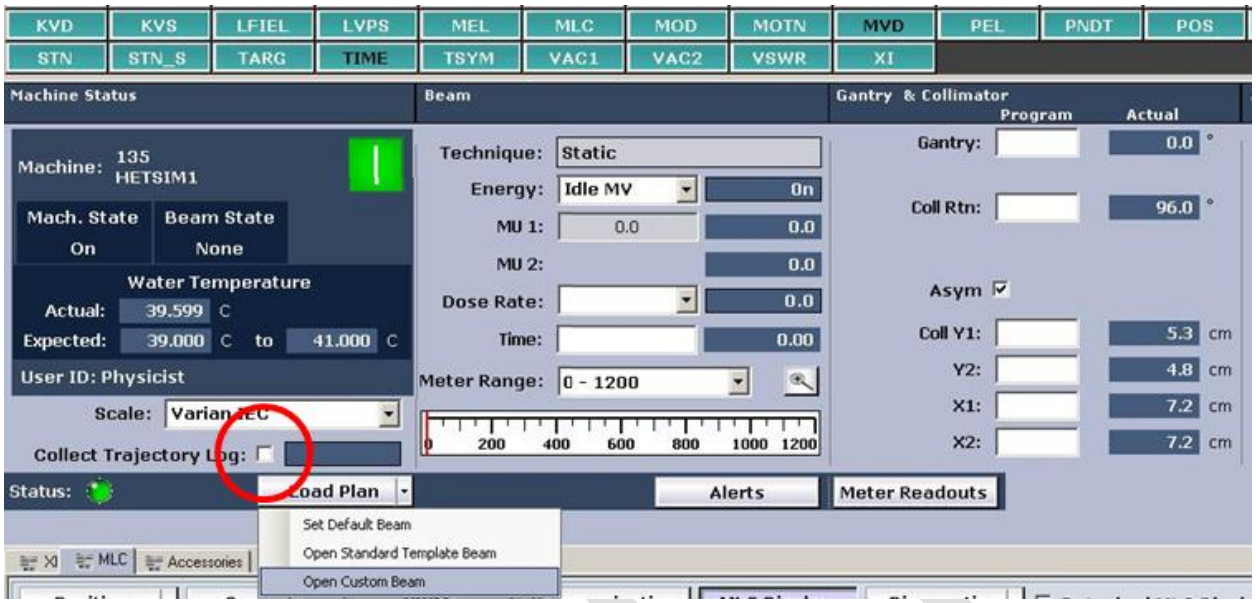
If motion on a moving axis stops or is not responding to motion, velocity or acceleration commands as expected by its motion model, then beam delivery is halted and a fault is generated.

In general, during beam delivery (including beamless segments) the control system continuously logs the radiation delivered and position of every moving axis every 10 msec or 20 msec (time interval depends on the control system version). The developer mode user can request that the these MU/position records be permanently stored on disk for later analysis.

Enabling Trajectory Logs

The developer mode user can request that a trajectory log be stored on disk for every beam delivered via Developer Mode. Trajectory logs are enabled by checking the box shown below on the Developer Mode user interface.

DRAFT



The box is not checked by default so it must be checked upon entering Developer Mode if trajectory logging is desired. While the box remains checked, the control system will generate a separate trajectory log for every beam delivered. A message to the user is displayed every time a trajectory log is generated, indicating the file name of the trajectory log. The file name contains the date and time that the Trajectory Log was generated. Trajectory logs are stored in the directory: C:\VMSOS\AppData\TDS\Output\TrajectoryLog\Research

Trajectory logs are currently stored in binary format. The exact format of trajectory record files is described in a separate document.

Trajectory logs can be large[8] and thus can potentially end up occupying a significant amount of disk space. It is recommended that trajectory logs be copied to another location at regular time intervals (*e.g.* every day, or after about a couple of hours of beam time) and then removed from the control system console computer.

Limitations

Interrupted Beams: Interrupted dynamic XML Beams pose a challenge in the current version of Developer Mode. If delivery of a dynamic Beam is interrupted because of a beam off or fault, then any moving axes will typically stop some small distance away from their required resumption point (overshoot). The amount of overshoot depends on the axis inertia. If the overshoot is greater than the internal axis tolerance then, an interlock (axis position interlock) will prevent immediate resumption of Beam delivery. Unfortunately, there is no indication on the Developer Mode interface as to what those

[8] On version 1.5 and 1.6 control systems trajectory logs require about one GB of disk space for every hour of recorded beam time.

resumption positions are and there is also no automatic way to instruct the system to move to those resumption positions. The end result is that, typically, in this situation, the XML beam must be reloaded and restarted from the beginning. This is especially true if any axis is intentionally moved by the user even further away from the resumption point.

Notice how control system response to an interrupted beam (whether interrupted due to interlock, fault or beam-off button) is different than interruption due to gating. In gating, when the gate opens the control system actively drives all moving axes (back) to the appropriate resumption point. In beams interrupted by interlock, fault or beam-off button the control system simply stops all moving axes in the shortest possible distance. That typically results in a (small) position overshoot relative to the point where the beam was interrupted.

Partial Beams (not supported): Partial Beams are not supported in the current version of Developer Mode. In the clinical Treatment Modes it is possible to instruct the system to deliver only the last X MU of a beam initially intended to deliver a total of Y MU ($X \leq Y$). The purpose of this feature in the Treatment Modes is to address the need of delivering, at a later time, the remainder of beams that had to be abandoned before completion. However, this feature is not supported in the current version of Developer Mode.

Dry Runs (not supported): The TrueBeam clinical Treatment modes support the concept of a "Dry Run" *i.e.* the ability to simulate all the dynamic motions of a Beam without actually delivering any MU [9]. However the current version of Developer Mode does not support Dry Runs.

CBCT Reconstructor not available in Developer Mode: The CBCT reconstructor that is typically available in Treatment Modes, is not currently connected to Developer Mode. Therefore, while XML Beams delivered through Developer Mode may acquire CBCT projections in any one of the imaging modes available in the Treatment modes, reconstruction must be performed through other means.

Maximum number of control points is 5000: The maximum number of control points that can be specified in the current version of Developer Mode is 5000.

Images taken by an XML beam may not be available in real time in their designated directories while the XML beam is still executing. The images typically become available a fraction of a second after delivery of the XML Beam completes.

Motion-only segments may exhibit unsynchronized motion: If a motion-only segment moves more than one axis, then the implied linearity between motions (i.e. all axes move proportionately) is maintained, unless the segment also moves the MLC. If a Motion only segment delivers neither MV nor KV radiation and there is MLC motion in the segment then the motions are not synchronized. In this non synchronized mode of motion,the axes do not follow the proportional linear motion implied in the general model and thus each axis moves independently, typically at maximum speed. In this

[9] In Treatment Modes, these Dry Runs can be performed inside the treatment room using the in-room pendant, since the primary use of this feature is to visually check equipment and patient clearances with respect to any dynamic motions to be performed during the treatment. Note that while here we are describing clinical mode features, Developer Mode is not intended for clinical use. As a reminder: Developer Mode is intended for non-clinical use only and is NOT cleared for use on humans.

uncoordinated motion case, the motion path is unpredictable as each axis is simoultaneously but independently moved between the positions specified at the two control points that define the segment.

Motion-only segments trigger instantaneous stops: When the control system transitions from a beam segment (a segment delivering MU) to a motion only segment and vice versa, the control system performs a synchronized deceleration of all moving axes and MU delivery until the motions and beam come to an instantaneous complete stop. Then motions and beam are immediately re-accelerated to the speeds and dose rate required in the next segment. While this behavior does not affect the dosimetry of the beam (since positions and beam are always accelerated and decelerated in a precisely synchronized fashion) the overall time required to deliver the beam may be extended if the beam contains a large number of beam to motion only segment transitions. The total beam time extension amounts to a small fraction of a second for each beam to motion only segment transition and vice versa.

In some versions of Developer Mode, the last control point in a beam must contain an MU specification even if the last segment delivers no MU (*i.e.* the MU specified in the last control point is the same as the MU specified in the next to last control point, or an MU counter value established at an earlier control point). This is simply a syntax peculiarity of some earlier versions of Developer Mode which otherwise has no dosimetric implications.

DRAFT

APPENDIX A: The Varian TrueBeam XML Schema

Following is the Varian TrueBeam XML Schema:

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:VMSHET="xsd.os.varian.com/HET" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <!--*****************************************************************************-->
    <!--    S E T   B E A M                                                          -->
    <!--    The root element defining the beam                                       -->
    <!--*****************************************************************************-->
    <xs:element name="SetBeam">
        <xs:complexType>
            <xs:all>
                <!--*****************************************************************************-->
                <!--    Specify The Version of the Varian Research Beam Schema (optional but
                        strongly recommended)
                        If the version is missing .....                                  -->
                <!--*****************************************************************************-->
                <xs:element name="SchemaVersion" type="xs:string" minOccurs="0" maxOccurs="1" />

                <!--*****************************************************************************-->
                <!--    Specify an integer number to be be used as a unique identifier for this-->
                <!--    Research Beam.                                                   -->
                <!--*****************************************************************************-->
                <xs:element name="Id" type="xs:int" minOccurs="1" maxOccurs="1" />

                <!--*****************************************************************************-->
                <!--    Specify a String identifier for this Research beam (optional).         -->
                <!--*****************************************************************************-->

                <xs:element name="GUID" type="xs:string" minOccurs="0" maxOccurs="1" />

                <!--*****************************************************************************-->
                <!-- Specify the MLC model required for this ResearchBeam (Optional)           -->
                <!-- Rule: If MLC model is specified, machine will accept experiment plan      -->
                <!-- only if equipped with same MLC model                                    -->
                <!--*****************************************************************************-->
                <xs:element name="MLCModel" minOccurs="0" maxOccurs="1">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <!--Three types of MLCs are possible: 80 leaf, 120 leaf and 120leaf high
                                                                              definition -->
                      <xs:enumeration value="NDS80" />
                      <xs:enumeration value="NDS120" />
                      <xs:enumeration value="NDS120HD" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>

                <!--*****************************************************************************-->
                <!-- Specify whether to track this beam and the tracking parameters (optional)
```

```
     Define the tracking enable status plus the tracking capability ratio.
     Tracking capability ratio range between 0 and 1.0, with 1.0 means tracking
     trajectory would
     be generated with full capabilities of axes.                              -->
<!--*************************************************************************-->
<xs:element name="Tracking" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <!--units are 1 deg/sec or 1 cm/sec ... -->
      <xs:element name="TrackingEnabled" type="xs:boolean" minOccurs="1"
                                                                  maxOccurs="1" />

      <!--*************************************************************************
      Capability ratio is restricted to be between 0.0 and 1.0
      *************************************************************************-->
      <xs:element name="CapabilityRatio" minOccurs="1" maxOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:double">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="1"/>
          </xs:restriction>
        </xs:simpleType>


      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--*************************************************************************
*************************************************************************
     Specify a tollerance table to be used (optional)
        RULES:
          1. If an individual axis has a specified zero tolerance value or
             a missing tolerance element, the Control System will impose its
             tight internal tolerance on that axis.
          2. If the entire tolerance table is missing, the Control System will
             impose its tight internal tolerance on all axes.
*************************************************************************-->
<xs:element name="TolTable" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>
      Tolerance Table
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <!--units are deg or cm ... -->
      <xs:element name="GantryRtn" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CollRtn" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchVrt" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchLat" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchLng" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchRtn" type="xs:double" minOccurs="0" maxOccurs="1" />
```

DRAFT

```
      <xs:element name="Y12" type="xs:double" minOccurs="0" maxOccurs="1" />
      <!-- ... independent Y tolerance -->
      <xs:element name="X12" type="xs:double" minOccurs="0" maxOccurs="1" />
      <!-- ... independent X tolerance -->
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--*************************************************************************-->
<!--Velocity limit table.
    Two consecutive control points define a segment.
    There is always a limit to how fast a segment can be delivered.
    Depending on MU delivered and motions performed in each segment, the
    limiting factor to how fast a segment can be delivered, turns out to be either
    the max dose rate or one of the moving axis.
    This is the "speed limiting parameter" for the segment which can be either
    one of the mechanical axes or the dose (rate).
    Obviously, the "limiting parameter" can change from segment to segment (e.g it
    can be dose rate in one segment, max gantry rotational speed in the next
    segment, leaf speed in the next etc.
    The control system maximizes the speed of the limiting parameter and then
    adjusts all other parameters proportionately (dose rate and speeds of moving
    axes) so that the linear relationship of dose vs position defined by the
    control points is followed precisely.
    If you want to limit the speed of any axis so that it never exceeds a certain
    speed in any one segment in your beam then you can use this table to set axis
    velocity limits. The maximum velocity that you specify will act as a velocity
    ceiling. The actual speed of any axis in any one segment may be lower if
    another axis (or does rate) is the limiting parameter.
    If the limit you specify is higher than the inherrent maximum axis velocity
    then your limit will be ignored and the intrinsic maximum velocity will be
    used a s a limit.
*************************************************************************-->
<xs:element name="VelTable" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>
      Max Velocity Table:
   </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <!--units are in degrees/sec or cm/sec ... -->
      <xs:element name="GantryRtn" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CollRtn" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchVrt" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchLat" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchLng" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="CouchRtn" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="X1" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="X2" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="Y1" type="xs:double" minOccurs="0" maxOccurs="1" />
      <xs:element name="Y2" type="xs:double" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
```

DRAFT

```
</xs:element>

<!--**************************************************************************
Specify any accessories to be used.
RULES:
1. A missing element for a slot means that ANY accessory can be installed in the
   slot without causing an interlock; i.e. "don't care".
2. The value '0' (zero) in a slot means that no accessory must be installed in
   the slot.
3. Any other integer value in a slot means that an accessory with that accessory
   code must be installed in that slot.
**************************************************************************-->
<xs:element name="Accs" minOccurs="1" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>
      The planned Accessory Codes for each slot.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Acc1" type="xs:int" minOccurs="0" maxOccurs="1" />
      <xs:element name="Acc2" type="xs:int" minOccurs="0" maxOccurs="1" />
      <xs:element name="Acc3" type="xs:int" minOccurs="0" maxOccurs="1" />
      <xs:element name="Acc4" type="xs:int" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--**************************************************************************
Control points list.
This is where you define the the dose vs. position relationship (trajectory
in dose/position space) describing the beam from beginning to end.
Typically at each control point the amount of dose and motions to be performed
are listed
**************************************************************************-->
<xs:element name="ControlPoints" minOccurs="1" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <!--*****************************************************************-->
      <!--The control points (2 is the minimum number of control points).      -->
      <!--*****************************************************************-->
      <!--General Control Point Rules:
            1) Unspecified (i.e. unplanned) STATIC Axes,
                 should NOT APPEAR in ANY Control Point. The Control System will
                 not enforce any tolerance restrictions on such axes.  However,
                 the Control System will still ensure that these axes do not move
                 from wherever they are placed once when the system transitions
                 to READY.
                 NOTE: MLC may be unspecified STATIC by this rule.
            2) Specified STATIC Axes, whose initial positions are contained in
                 the Beam need only appear in the First Control Point. The
                 controller will ensure that they do not move from wherever they
                 are placed once it goes to READY.
            3) Specified DYNAMIC Axes, whose position "program" is contained in
```

```
                the Beam need only appear in the First Control Point and any
                subsequent Control Point where there is a change in their
                position.  The controller will ensure that they follow their
                position "program" during the treatment).
            4) Only one independent jaw of a pair may be appear in a Control
                Point. It is not necessary for both to appear. This could occur
                if only one jaw is dynamic.
            5) MU is cumulative. From one control point to the next, it can only
                go up or stay the same. It cannot decrease.
            6) There are some special additional rules for the first control
                    point. The first control point:
                    A) Must specify a Dose value of 0 MU
                    B) Must contain "Mu" = 0.
                    C) Must contain "DRate".
                    D) Must contain "SubBeam"
             7) Subsequent Control Points:
                    A) Must contain "Energy" only if it is changing
                    B) Must contain "Mu" only if it is changing (i.e. not a "no
                       dose" segment).
                    C) Must contain "DRate" only if it is changing
                    D) Must contain axes whose positions are changing (dynamic
                       axes).
                    E) May not contain axes (including MLC) which were not
                       contained in the first Control Point.
    -->
    <!--*****************************************************************-->
    <xs:element name="Cp" minOccurs="2" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
            for more details.)
        </xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="TreatProgressEvent" type="xs:string" minOccurs="0"
                                                                 maxOccurs="1" />

          <!--*****************************************************************-->
          <!--This is an optional tag. A SubBaeam element that is used to tag
              the control point as the beginning of a new "Sub" Beam -->
          <!--*****************************************************************-->
          <xs:element name="SubBeam" minOccurs="0" maxOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Seq" type="xs:unsignedInt" minOccurs="1"
                                                                 maxOccurs="1" />
                <xs:element name="SubbeamGUID" type="xs:string" minOccurs="0"
                                                                 maxOccurs="1" />
                <xs:element name="Name" type="xs:string" minOccurs="0"
                                                                 maxOccurs="1" />
                <xs:element name="MaxRadTime" type="xs:double" minOccurs="0"
                                                                 maxOccurs="1" />
                <xs:element name="TrackingTrainingOnly" type="xs:boolean"
                                                    minOccurs="0" maxOccurs="1" />
```

```
    </xs:sequence>
  </xs:complexType>
</xs:element> <!--END SubBeam-->

<!--*************************************************************-->
<!--1) Energy starting at this control point - A unique signature for
       the energy in the form "dds" where
          dd = 0-99,
          s = 'x', 'e', or 'h', where
                x : MV X-Rays
                e : MeV electrons
                h : Mev HDTSe- electons.
                k : KV beams
                Examples: "6x" (6 MV X-Rays) and "12e" (12 Mev
                electrons) and "0k" (kv beam)
-->
<!--*************************************************************-->
<xs:element name="Energy" type="xs:string" minOccurs="0"
                                                    maxOccurs="1" />

<!--*****************************************************************
Cumulative Dose in units of 1 MU (floating point number, can thus be
fractional MU)
Since dose is cumulative, the dose for this control point must be
greater or equal to the dose specified in the previous control point.
This dose specified in this control point minus the dose specified in
the previous control point is the dose delivered in the last segment
(segment defined by this control point and previous control point
If dose is the same as previous control point then the segment ending
at this control point delivers no dose (no dose segment)
*****************************************************************-->
<xs:element name="Mu" minOccurs="0" maxOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:double">
      <!--Obviously dose cannot be less than 0-->
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<!--*****************************************************************
Dose rate in MU per minute for this control point
Note: The dose rate specified here acts as a dose rate "ceiling", the
actual dose rate is computed on a per segment basis and may be lower
than this dose rate if another moving axis is the speed limiting
parameter (SLP) -->

<xs:element name="DRate" minOccurs="0" maxOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:double">
      <!--Obviously dose rate cannot be less than 0-->
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
```

```
    </xs:element>

    <!-- Axis positions. Units are degrees for rotational axes and
         centimeters for linear axes. -->
    <!-- In Varian internal scale.                                    -->
    <xs:element name="GantryRtn" type="xs:double" minOccurs="0"
                                                        maxOccurs="1" />
    <xs:element name="CollRtn" type="xs:double" minOccurs="0"
                                                        maxOccurs="1" />
    <xs:element name="CouchVrt" type="xs:double" minOccurs="0"
                                                        maxOccurs="1" />
    <xs:element name="CouchLat" type="xs:double" minOccurs="0"
                                                        maxOccurs="1" />
    <xs:element name="CouchLng" type="xs:double" minOccurs="0"
                                                        maxOccurs="1" />
    <xs:element name="CouchRtn" type="xs:double" minOccurs="0"
                                                        maxOccurs="1" />

    <!--Note: The Collimator positions, (including MLC) designate the
        resulting field size at isocenter -->
    <xs:element name="Y1" type="xs:double" minOccurs="0" maxOccurs="1" />
    <xs:element name="Y2" type="xs:double" minOccurs="0" maxOccurs="1" />
    <xs:element name="X1" type="xs:double" minOccurs="0" maxOccurs="1" />
    <xs:element name="X2" type="xs:double" minOccurs="0" maxOccurs="1" />

    <!--*******************************************
    *  MLC positions in cm (projected at isocenter)  *
    *******************************************-->
    <xs:element name="Mlc" minOccurs="0" maxOccurs="2">
      <xs:complexType>
        <xs:sequence>
          <!-- ID = 1 means primary MLC, ID = 2 means secondary MLC -->
          <xs:element name="ID" type="xs:int" minOccurs="1"
                                                    maxOccurs="1" />

          <!-- Array of MLC positions for Bank B for this control point -
               leaf positions are delimited by spaces.
               The first leaf in the array is B-1, and the last leaf is
               B-40, B-60,or A-26. The positions are defined in the array
               as double, in 1 cm units, in Varian scale.
               Values represent logical (as opposed to physical) positions
               in the Isocenter Plane.
          -->
          <xs:element name="B" type="xs:string" minOccurs="1"
                                                    maxOccurs="1" />

          <!-- Same as above but for Bank A -->
          <xs:element name="A" type="xs:string" minOccurs="1"
                                                    maxOccurs="1" />
        </xs:sequence>
      </xs:complexType> <!--END of MLC Positions. -->
    </xs:element>

  </xs:sequence>  <!--***END OF Cp elements list***-->
```

DRAFT

```
        </xs:complexType>
      </xs:element> <!--***END OF Cp***-->
    </xs:sequence>
  </xs:complexType>
</xs:element>  <!--***END OF ControlPoints (list)***-->

<!--*****************************************************************************
*****************************************************************************
IMAGING PARAMETERS (Optional)
*****************************************************************************
*****************************************************************************-->
<xs:element name="ImagingParameters" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <!--*************************************************************************
      *  Specify whether images will be taken during MV beam or outside MV       *
      *  irradiation                                                             *
      *************************************************************************-->
      <xs:choice minOccurs="1" maxOccurs="1">

        <!--*********************************************************************
        Type for imaging during treatment.

        Images may be acquired during treatment.
        If the beam is paused while XI is in an acquiring state, the current
        image(s) will be dropped and the treatment beam can be resumed under SPV
        control at the control point where it was paused.

        If the system is equipped for imaging, but no images have to be acquired
        for a treatment beam, this type is used at well. In this case, imaging
        points shall contain the initial arm positions, but no acquisitions
        *********************************************************************-->
        <xs:element name="DuringTreatment">
          <xs:complexType>
            <xs:sequence></xs:sequence>
          </xs:complexType>
        </xs:element>

        <!--*********************************************************************
        Type for imaging outside treatment.

        Supervisor shall terminate the beam as soon as one of the following
        condition is fulfilled:
        1. All acquisitions are completed (normal termination).
        2. The maximum MU limit (MaxMu) is reached.
        3. The operator stops the beam.
        4. A fault occurs.
        Supervisor beam state never changes to 'paused'.
        The reason of termination can be determined from the supervisor beam
        history.

        Note:
        The corresponding beam has the following characteristics.
        1. There may be no MV energy defined (e.g. kV imaging or dark field
```

DRAFT

```
      acquisition),
      MaxMu shall be 0 in this case.
    2. The meterset values in the control points have no semantics.
    3. There may be 2 successive control points with no changes in any value.
    ************************************************************************-->
    <xs:element name="OutsideTreatment">

      <xs:complexType >
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" name="MaxMu"
                                                          type="xs:double" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

  </xs:choice>   <!--***END OF either During or Outside Irradiation
                                                              imaging***-->

  <!--************************************************************************
  *  Specify Custom Target Permission (true/false)                          *
  ************************************************************************-->
  <xs:element minOccurs="0" maxOccurs="1" name="CustomTargetPermission"
                                        type="xs:boolean" default="false"/>

  <!--************************************************************************
  Max Velocity Table:
  This table can be used to specify top velocities of axes to be used in
  the treatment.

  RULES:
  1. If an individual axis has a specified max velocity value, then the
     Control System will use the smaller of the specified max velocity and the
     maximum supported max velocity of that axis.
  ************************************************************************-->
  <xs:element minOccurs="0" maxOccurs="1" name="ImagingVelTable">
    <xs:complexType>
      <xs:sequence>
        <!--List of max velocities desired for each Clinac axis -->
        <!--units are 1 deg/sec or 1 cm/sec ... -->
        <xs:element name="GantryRtn" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="CollRtn" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="CouchVrt" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="CouchLat" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="CouchLng" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="CouchRtn" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="X1" type="xs:double" minOccurs="0"
                                                          maxOccurs="1" />
        <xs:element name="X2" type="xs:double" minOccurs="0"
```

DRAFT

```
                                                                maxOccurs="1" />
      <xs:element name="Y1" type="xs:double" minOccurs="0"
                                                                maxOccurs="1" />
      <xs:element name="Y2" type="xs:double" minOccurs="0"
                                                                maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element> <!--END OF Imaging Velocity Table -->



<xs:element minOccurs="0" maxOccurs="1" name="LatchBEL" type="xs:boolean"
                                                                default="true"/>

<xs:element minOccurs="0" maxOccurs="1" name="LatchKVBEL" type="xs:boolean"
                                                                default="true"/>

<!--*********************************************************************
Sequence of imaging points.
An initial imaging point at Cp=0 is mandatory for all imaging axes, which
are specified in a subsequent imaging point.
*********************************************************************-->
<xs:element minOccurs="1" maxOccurs="1" name="ImagingPoints" >
  <xs:complexType>
    <xs:sequence>
      <!--*****************************************************************
      Imaging point.
      Refers to fractional control point indices, e.g. 3.5 means with all
      Clinac axes averaged between control points with indices 3 and 4.
      In case of a beam group (or superbeam), Cp refers to the fractional
      control point index of the combined beam.
      Example:
      Beam group with 3 beams.
      First beam has 5 control points:        0 &#8804; Cp &#8804; 4.
      Second beam has 2 control points:       5 &#8804; Cp &#8804; 6.
      Third beam has 3 control points:        7 &#8804; Cp &#8804; 9.

      *****************************************************************-->
      <xs:element minOccurs="1" maxOccurs="unbounded" name="ImagingPoint">
        <xs:complexType>
          <xs:sequence>
            <!--*************************************************************
            The fractional control point. This refers to the list of beam
            control points in element ControlPoints above.
            This factional control point is essentially what determines
            where in the beam the image will be taken. In other words, this
            fractional control point i what corelates the Imaging and beam
            control points.
            *************************************************************-->
            <xs:element minOccurs="1" maxOccurs="1" name="Cp"
                                                          type="xs:double" />

            <!--*************************************************************
            Acquisitions at this imaging point. You can either do a simple
```

```
acquisition or start/stop a continuous acquisition at any
imaging control point.
**************************************************************-->
<xs:element minOccurs="0" maxOccurs="8" name="Acquisition"
                                              type="Acquisition" />
<xs:element minOccurs="0" maxOccurs="8" name="AcquisitionStart"
                                              type="Acquisition" />
<xs:element minOccurs="0" maxOccurs="8" name="AcquisitionStop"
                                              type="Acquisition" />

<!--**************************************************************
kV filter settings (optional) at this imaging point
**************************************************************-->
<xs:element minOccurs="0" maxOccurs="1" name="KvFilters">
  <xs:complexType>
    <xs:all>
      <xs:element name="Shape" type="KvFilterType" minOccurs="0"
                                                   maxOccurs="1" />
      <xs:element name="Foil" type="KvFilterType" minOccurs="0"
                                                   maxOccurs="1" />
    </xs:all>
  </xs:complexType>
</xs:element>

<!--**************************************************************
desired (optional) arm /kv blade positions before the image
acquisition. Specify either blade tracking or blade positions
**************************************************************-->
<xs:element name="KvBlades" minOccurs="0" maxOccurs="1">
  <xs:complexType >
    <xs:sequence>
      <xs:choice minOccurs="1" maxOccurs="1">
        <xs:element name="Tracking" type="xs:boolean" />
        <!--**************************************************
        Positions of the kV Blades; units in cm
        **************************************************-->
        <xs:element name="Positions">
          <xs:complexType>
            <xs:all> <!--Must specify all kV collimator
                          positions (but unlike MV collimators
                              they can be listed in any order)-->
              <xs:element minOccurs="1" maxOccurs="1"
                                  name="KVX1" type="DoubleNone" />
              <xs:element minOccurs="1" maxOccurs="1"
                                  name="KVX2" type="DoubleNone" />
              <xs:element minOccurs="1" maxOccurs="1"
                                  name="KVY1" type="DoubleNone" />
              <xs:element minOccurs="1" maxOccurs="1"
                                  name="KVY2" type="DoubleNone" />
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
```

```
            </xs:complexType>
          </xs:element>


          <xs:element minOccurs="0" maxOccurs="1" name="Mvd"
                                                  type="ArmPositionsType" />
          <xs:element minOccurs="0" maxOccurs="1" name="Kvd"
                                                  type="ArmPositionsType" />
          <xs:element minOccurs="0" maxOccurs="1" name="Kvs"
                                                  type="ArmPositionsType" />


          <!--desired arm position after the image acquisition -->
          <xs:element minOccurs="0" maxOccurs="1" name="MvdAfter"
                                                  type="ArmPositionsType" />
          <xs:element minOccurs="0" maxOccurs="1" name="KvdAfter"
                                                  type="ArmPositionsType" />
          <xs:element minOccurs="0" maxOccurs="1" name="KvsAfter"
                                                  type="ArmPositionsType" />
        </xs:sequence>
      </xs:complexType>
    </xs:element> <!--END OF single imaging point element-->
  </xs:sequence>
</xs:complexType>
</xs:element> <!--END OF Imaging Points sequence -->

<!--**************************************************************************
Tolerances of the imaging equipment (PU arms, kV collimation system).
**************************************************************************-->
<xs:element minOccurs="1" maxOccurs="1" name="ImagingTolerances">
  <xs:complexType>
    <xs:sequence>
      <!--units are deg or cm -->
      <!--**********************************************************
      Tolerances for MV detector, kV detector and kV source arms.
      **********************************************************-->
      <xs:element minOccurs="0" maxOccurs="1" name="Mvd"
                                                type="ArmTolerances" />
      <xs:element minOccurs="0" maxOccurs="1" name="Kvd"
                                                type="ArmTolerances" />
      <xs:element minOccurs="0" maxOccurs="1" name="Kvs"
                                                type="ArmTolerances" />
      <!--**********************************************************
      kV Blade (kV collimator) tollerances
      **********************************************************-->
      <xs:element minOccurs="0" maxOccurs="1" name="KvBlades">
        <xs:complexType>
          <xs:sequence>
            <!--units are deg or cm-->
            <xs:element name="KVY1" type="xs:double" minOccurs="1"
                                                     maxOccurs="1" />
            <xs:element name="KVY2" type="xs:double" minOccurs="1"
                                                     maxOccurs="1" />
            <xs:element name="KVX1" type="xs:double" minOccurs="1"
```

DRAFT

```
                                                                maxOccurs="1" />
            <xs:element name="KVX2" type="xs:double" minOccurs="1"
                                                                maxOccurs="1" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>  <!--***END OF ImagingTolerances***-->

<!--**************************************************************************
Gating Parameters. If the respiratory gating that feeds into the XI is to
be used then specify gating parameters here
**************************************************************************-->
<xs:element minOccurs="0" maxOccurs="1" name="GatingParameters">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Source"
                                                        type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Filter"
                                                        type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="QualityThreshold"
                                                        type="xs:double" />
      <xs:element minOccurs="0" maxOccurs="1" name="KVGating"
                                                       type="xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="Position">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="X"
                                                        type="xs:double" />
            <xs:element minOccurs="1" maxOccurs="1" name="Y"
                                                        type="xs:double" />
            <xs:element minOccurs="1" maxOccurs="1" name="Z"
                                                        type="xs:double" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>  <!--***END OF Position***-->

      <xs:element minOccurs="0" maxOccurs="1" name="Orientation">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="Qx"
                                                        type="xs:double" />
            <xs:element minOccurs="1" maxOccurs="1" name="Qy"
                                                        type="xs:double" />
            <xs:element minOccurs="1" maxOccurs="1" name="Qz"
                                                        type="xs:double" />
            <xs:element minOccurs="1" maxOccurs="1" name="Q0"
                                                        type="xs:double" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>  <!--***END OF Orientation***-->

      <xs:element minOccurs="1" maxOccurs="unbounded" name="GatingWindow">
```

DRAFT

```
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="1" name="Axis"
                                                              type="xs:string" />
                    <xs:element minOccurs="1" maxOccurs="1" name="Entry"
                                                              type="xs:double" />
                    <xs:element minOccurs="1" maxOccurs="1" name="Exit"
                                                              type="xs:double" />
                    <xs:element minOccurs="0" maxOccurs="1" name="EntryDelay"
                                                              type="xs:double" />
                    <xs:element minOccurs="0" maxOccurs="1" name="FaultOnExit"
                                                             type="xs:boolean" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>  <!--***END OF GatingWindow***-->

              <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="MVAcquisitionTrigger">
                  <xs:complexType>
                    <xs:complexContent>
                      <xs:extension base="AcquisitionTrigger" />
                    </xs:complexContent>
                  </xs:complexType>
                </xs:element>  <!--***END OF MVAcquisitionTrigger***-->

                <xs:element name="KVAcquisitionTrigger">
                  <xs:complexType>
                    <xs:complexContent>
                      <xs:extension base="AcquisitionTrigger" />
                    </xs:complexContent>
                  </xs:complexType>
                </xs:element>  <!--***END OF KVAcquisitionTrigger***-->

              </xs:choice>
            </xs:sequence>
          </xs:complexType>
        </xs:element>   <!--***END OF GatingParameters***-->

      </xs:sequence>
    </xs:complexType>
  </xs:element>   <!--***END OF ImagingParameters***-->


  <!--*******************************************
  BEAM HOLD DEVICES(Optional)
  BeamHoldDevice Definition
  The planned beam hold devices.

  RULES:
  1. A missing beam hold device means that the beam hold devie must not be ready.
  2. A present beam hold device must be ready
  3. The MV beam gets held if the beam hold device requests to held the beam.
  4. Id attribute specifies the id of the beam hold device (number between 0 and 15)
  *******************************************-->
```

```
            <xs:element name="BeamHoldDevices" minOccurs="0" maxOccurs="1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Dev" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                      <xs:attribute name="Id" type="xs:int" />
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>

        </xs:all>
    </xs:complexType>
</xs:element> <!--END OF SetBeam element -->


<!--**************************************************************************************
**************************************************************************************
T Y P E     D E F I N I T I O N S
**************************************************************************************
**************************************************************************************-->


<!--**************************************************************************************
TypeDef for container: "Acquisition" describing one image source.
**************************************************************************************-->
<xs:complexType name="Acquisition">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="AcquisitionId" type="xs:int" />

    <!--**************************************************************************
    Container for any detailed XI acquisition information needed by SPV
    in setup phase.
    Handshake: hand shaking between SPV and XI required
      If yes SPV will pause delivery until XI is ready for acquisition
      If not SPV will merely trigger the acquisition in time
    KV: is kv Beam
    MVDose: estimated total amount of MVDose required in order to acquire the image
      This is used by SPV to calculate radiation time.
    Gating: for gated MV radshot
    **************************************************************************-->
    <xs:element minOccurs="1" maxOccurs="1" name="AcquisitionSpecs">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="Handshake" type="xs:boolean" />
          <xs:element minOccurs="0" maxOccurs="1" name="KV" type="xs:boolean" />
          <xs:element minOccurs="0" maxOccurs="1" name="MVDose" type="xs:double" />
          <!--**********************************************************************
          Enumeration for gated MV radshot: none, once or continuously
          **********************************************************************-->
            <!--*****************MAX LINE*****************************************-->
          <xs:element minOccurs="0" maxOccurs="1" name="Gating">
            <xs:simpleType>
```

```
          <xs:restriction base="xs:string">
            <xs:enumeration value="none" />
            <xs:enumeration value="once" />
            <xs:enumeration value="continuously" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element> <!--END OF AcquisitionSpecs -->
<!--*************************************************************************************
Container for image acquisition specifications
*************************************************************************************-->
<xs:element minOccurs="0" maxOccurs="1" name="AcquisitionParameters">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="ImageMode">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="Overwrite">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <xs:attribute name="parameter" type="xs:string" use="required" />
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:string" use="required" /> <!--ImageMode has
                                                        a mandatory attribute "id"-->
        </xs:complexType>
      </xs:element> <!--END OF ImageMode -->

      <xs:element minOccurs="0" maxOccurs="1" name="AcquisitionMode">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="Overwrite">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <xs:attribute name="parameter" type="xs:string" use="required" />
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:string" use="required" /> <!--AcquisitionMode
                                                    has a mandatory attribute "id"-->
        </xs:complexType>
      </xs:element> <!--END OF AcquisitionMode -->


      <xs:element minOccurs="0" maxOccurs="1" name="CalibrationSet" type="xs:string" />
```

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="ImageDestination"
                                                          type="xs:string" />
<xs:element minOccurs="0" maxOccurs="unbounded" name="NotificationDestination"
                                                          type="xs:string" />
<xs:element minOccurs="0" maxOccurs="1" name="Movie">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Destination" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="width" type="xs:int" />
            <xs:element minOccurs="0" maxOccurs="1" name="height" type="xs:int" />
            <xs:element minOccurs="0" maxOccurs="1" name="frameRate"
                                                          type="xs:double" />
            <xs:element minOccurs="0" maxOccurs="1" name="crop" type="xs:boolean" />
            <xs:element minOccurs="0" maxOccurs="1" name="invert"
                                                        type="xs:boolean" />
            <xs:element minOccurs="0" maxOccurs="1" name="pixelValue0"
                                                             type="xs:int" />
            <xs:element minOccurs="0" maxOccurs="1" name="pixelValue255"
                                                          type="xs:int" />
            <xs:element minOccurs="0" maxOccurs="1" name="cutOff"
                                                          type="xs:int" />
            <xs:element minOccurs="0" maxOccurs="1" name="weight"
                                                          type="xs:double" />
          </xs:sequence>
        </xs:complexType>
      </xs:element> <!--END OF (Movie) Parameters -->
    </xs:sequence>
  </xs:complexType>
</xs:element> <!--END OF Movie -->

<xs:element minOccurs="0" maxOccurs="1" name="RaiseFault" type="xs:boolean" />
<xs:element minOccurs="0" maxOccurs="1" name="Cache" type="xs:boolean" />
<xs:element minOccurs="0" maxOccurs="1" name="HistogramRoi">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="X1" type="xs:int" />
      <xs:element minOccurs="1" maxOccurs="1" name="Y1" type="xs:int" />
      <xs:element minOccurs="1" maxOccurs="1" name="X2" type="xs:int" />
      <xs:element minOccurs="1" maxOccurs="1" name="Y2" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
</xs:element> <!--END OF HistogramRoi -->

<!--*********************************************************
Specify either kV or MV parameters
*********************************************************-->
<xs:choice minOccurs="1" maxOccurs="1">
  <!--*********************************************************
  MV parameters are just energy and dose rate
  *********************************************************-->
  <xs:element name="MV">
```

DRAFT

```
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" maxOccurs="1" name="Energy" type="xs:string" />
                <xs:element minOccurs="0" maxOccurs="1" name="DoseRate" type="xs:double" />
              </xs:sequence>
            </xs:complexType>
          </xs:element> <!--END OF MV (parameters)-->

          <!--*********************************************************
          kV has more parameters
          *********************************************************-->
          <xs:element name="KV">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="1" maxOccurs="1" name="KiloVolts" type="xs:double" />
                <xs:element minOccurs="1" maxOccurs="1" name="MilliAmperes"
                                                                  type="xs:double" />
                <xs:element minOccurs="1" maxOccurs="1" name="MilliSeconds"
                                                                  type="xs:double" />
                <xs:element minOccurs="0" maxOccurs="1" name="eFocalSpot" default="Large">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="Small"/>
                      <xs:enumeration value="Large"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element> <!--END OF eFocalSpot-->

                <xs:element minOccurs="0" maxOccurs="1" name="eFluoroLevelControl"
                                                                      default="Low">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="None"/>
                      <xs:enumeration value="Low"/>
                      <xs:enumeration value="High"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element> <!--END OF eFluoroLevelControl-->

                <xs:element minOccurs="0" maxOccurs="1" name="AutoBrightnessControl"
                                                  type="xs:boolean" default="false" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>  <!--END OF kV (parameters)-->
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element> <!--END OF AcquisitionParameters -->
</xs:sequence>
</xs:complexType> <!--END OF  Acquisition -->

<!--*************************************************************************************
TypeDef for simple enumeration of the three kV filter positions 0,1,2
*************************************************************************************-->
```

```
<xs:simpleType name="KvFilterType">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="2" />
  </xs:restriction>
</xs:simpleType>

<!--*****************************************************************************
Type Element Definition for Arm positions
*****************************************************************************-->
<xs:complexType name="ArmPositionsType">
  <xs:choice>
    <!--*************************************************************************
    Predefined positions for PU:
    Position Type:
    retracted - arm is in a pre-defined fully retracted position.
    extended  - arm is either in a pre-defined extended position, or in a pre-defined
                clinical area (the VRT and LNG coordinates are within a pre-defined
                polygon where the image can be taken according to the treatment plan).
    mid       -  arm is in a pre-defined "Mid" position between retracted and extended.
    none      - all other positions. When used in setTarget, means cancel target.
    *************************************************************************-->

    <xs:element name="SpecialPosition" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="retracted" />
          <xs:enumeration value="mid" />
          <xs:enumeration value="extended" />
          <xs:enumeration value="none" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>

    <!--*************************************************************************
    Axes definitions for a working position of an arm.
    *************************************************************************-->
    <xs:element name="Positions" minOccurs="0" maxOccurs="1" >
      <xs:complexType>
        <xs:sequence>
          <!--units are deg or cm-->
          <xs:element minOccurs="1" maxOccurs="1" name="Lat" type="xs:double" />
          <xs:element minOccurs="1" maxOccurs="1" name="Lng" type="xs:double" />
          <xs:element minOccurs="1" maxOccurs="1" name="Vrt" type="xs:double" />
          <xs:element minOccurs="1" maxOccurs="1" name="Pitch" type="xs:double" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>


<!--*****************************************************************************
TypeDef for Tolerances of a PU arm.
```

DRAFT

```
    ********************************************************************************************-->
    <xs:complexType name="ArmTolerances">
      <xs:sequence>
        <!--units are deg or cm-->
        <xs:element name="Lat" type="xs:double" minOccurs="1" maxOccurs="1" />
        <xs:element name="Lng" type="xs:double" minOccurs="1" maxOccurs="1" />
        <xs:element name="Vrt" type="xs:double" minOccurs="1" maxOccurs="1" />
        <xs:element name="Pitch" type="xs:double" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>

    <!--********************************************************************************************
    TypeDef for Acquisition trigger
    ********************************************************************************************-->
    <xs:complexType name="AcquisitionTrigger">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="TriggerDelay" type="xs:double" />
        <xs:element minOccurs="0" maxOccurs="1" name="TriggerOnEnter" type="xs:boolean" />
        <xs:element minOccurs="0" maxOccurs="1" name="TriggerOnExit" type="xs:boolean" />
        <xs:element minOccurs="0" maxOccurs="1" name="SingleTrigger" type="xs:boolean" />
      </xs:sequence>
    </xs:complexType>


    <!--********************************************************************************************
    TypeDef for simple element "none" string
    ********************************************************************************************-->

    <xs:simpleType name="NoneValue">
      <xs:restriction base="xs:string">
        <xs:enumeration value="none" />
      </xs:restriction>
    </xs:simpleType>

    <!--********************************************************************************************
    TypeDef for simple element DoubleNone i.e.xs:double extended to include the value "none"
    ********************************************************************************************-->
    <xs:simpleType name="DoubleNone">
      <xs:union memberTypes="xs:double NoneValue  " />
    </xs:simpleType>

</xs:schema>
```

DRAFT

APPENDIX B: Accessories List

Accessory	Code (Range)	Slot	Applicator Type
NoAccy1	0	InterfaceMount	
NoAccy2	0	AccessoryMount	
NoAccy3	0	ElectronAperture	
NoAccy4	0	Compensator Mount	
UnkAcc1	4095	InterfaceMount	
UnkAcc2	4095	AccessoryMount	
UnkAcc3	4095	ElectronAperture	
UnkAcc4	4095	Compensator Mount	
UnkAcc5	127	ElectronAperture	
W15R30U	3920	InterfaceMount	
W15L30U	3921	InterfaceMount	
W15IN30U	3922	InterfaceMount	
W15OUT30U	3923	InterfaceMount	
W30R30U	3924	InterfaceMount	
W30L30U	3925	InterfaceMount	
W30IN30U	3926	InterfaceMount	
W30OUT30U	3927	InterfaceMount	
W45R20U	3912	InterfaceMount	
W45L20U	3913	InterfaceMount	
W45IN20U	3914	InterfaceMount	
W45OUT20U	3915	InterfaceMount	
W60R15U	3900	InterfaceMount	
W60L15U	3901	InterfaceMount	
W60IN15U	3902	InterfaceMount	
W60OUT15U	3903	InterfaceMount	
W15R30L	3920	Compensator Mount	
W15L30L	3921	Compensator Mount	
W15IN30L	3922	Compensator Mount	
W15OUT30L	3923	Compensator Mount	
W30R30L	3924	Compensator Mount	
W30L30L	3925	Compensator Mount	
W30IN30L	3926	Compensator Mount	
W30OUT30L	3927	Compensator Mount	
W45R20L	3912	Compensator Mount	
W45L20L	3913	Compensator Mount	
W45IN20L	3914	Compensator Mount	
W45OUT20L	3915	Compensator Mount	

DRAFT

W60R15L	3900	Compensator Mount	
W60L15L	3901	Compensator Mount	
W60IN15L	3902	Compensator Mount	
W60OUT15L	3903	Compensator Mount	
A06	4000	AccessoryMount	ELECTRON_SQUARE
A10	4001	AccessoryMount	ELECTRON_SQUARE
A15	4002	AccessoryMount	ELECTRON_SQUARE
A20	4003	AccessoryMount	ELECTRON_SQUARE
A25	4004	AccessoryMount	ELECTRON_SQUARE
A10X6	4006	AccessoryMount	ELECTRON_RECT
FT.POINTER	8167	InterfaceMount	
FFDA(A10+)	4094	ElectronAperture	
FFDA(A06)	126	ElectronAperture	
e-ARC Applicator	4094	AccessoryMount	
PF Grat Upper	3801	InterfaceMount	
PF Grat Lower	3792	Compensator Mount	
CustomBlockTray	1 - 1664	AccessoryMount	
UpperCustomComp	1 - 364	InterfaceMount	
LowerCustomComp	1 - 364	Compensator Mount	
CustomFFDA6	1 - 125	ElectronAperture	
CustomFFDA	3328 - 3707	ElectronAperture	
IsoCal Plate	3600	InterfaceMount	
CustomSRSAppl	3265 - 3326	AccessoryMount	STEREOTACTIC
CustomSRSMount	3264	InterfaceMount	

DRAFT